T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DERİN PEKİŞTİRMELİ ÖĞRENMEYE DAYALI OTONOM SÜRÜŞ İÇİN AÇIKLANABİLİR YAPAY ZEKA

YÜKSEK LİSANS TEZİ

MUHSİN KOMPAS

DENİZLİ, AĞUSTOS - 2024

T.C.
PAMUKKALE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DERİN PEKİŞTİRMELİ ÖĞRENMEYE DAYALI OTONOM SÜRÜŞ İÇİN AÇIKLANABİLİR YAPAY ZEKA

YÜKSEK LİSANS TEZİ

MUHSİN KOMPAS

DENİZLİ, AĞUSTOS - 2024

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde bilimsel etiğe ve akademik kurallara özenle riayet edildiğini; bu çalışmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etiğe uygun olarak kaynak gösterildiğini ve alıntı yapılan çalışmalara atfedildiğine beyan ederim.

MUHSİN KOMPAS

ÖZET

DERİN PEKİŞTİRMELİ ÖĞRENMEYE DAYALI OTONOM SÜRÜŞ İÇİN AÇIKLANABİLİR YAPAY ZEKA
YÜKSEK LİSANS TEZİ
MUHSİN KOMPAS
PAMUKKALE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
(TEZ DANIŞMANI:DR. ÖĞR. ÜYESİ İBRAHİM KÖK)

DENİZLİ, AĞUSTOS - 2024

Son yıllarda, yapay zeka ve bilgisayar teknolojilerindeki hızlı ilerlemenin de etkisiyle otonom araç teknolojisinde kayda değer bir artış yaşanmıştır. Günümüzde birçok büyük şirket ve endüstri paydaşı otonom araçlara yönelik aktif çalışmalar yürütmektedir. Otonom sürüş teknolojisinin veri odaklı ve derin öğrenme temelli tekniklerle hızla ilerlemesi, trafikteki sorunları azaltma ve verimliliği artırma potansiyeline sahip olsa da bu modellerin karmaşıklığı ve kara kutu yapısı nedeniyle karar verme süreçlerinin anlaşılabilirliği ve şeffaflığında önemli bir engel teşkil etmektedir. Bu bağlamda bir dizi hukuki, etik ve sosyo-psikolojik sorun ortaya çıkmaktadır. Açıklanabilir yapay zeka (Explainable Artificial Intelligence—XAI) metodolojilerinin uygulanması, otonom sürüş sistemlerinin güvenilirliğinin, hesap verebilirliğinin ve şeffaflığının arttırılmasında gelecek vaat eden çalışma alanı olarak ortaya çıkmıştır. Bu bağlamda XAI yasal düzenlemelere uygunluğun sağlanması, kullanıcıların güveninin arttırılması ve yazılım geliştirme tarafında eksikliklerin belirlenmesinde önemli bir rol oynama potansiyeline sahiptir. Bu tez çalışmasında, derin pekiştirmeli öğrenme (Deep Reinforcement Learning—DRL) tabanlı otonom sürüş sistemlerinin performansları ve XAI yaklaşımları incelenmiş, farklı senaryolar altında değerlendirilmiştir. Ayrıca XAI yöntemlerinin, otonom araçların yasal düzenlemelerle uyumlu hale getirilmesi ve toplum tarafından kabul edilmesini sağlama konusundaki rolü vurgulanmıştır. DRL modeli olarak TD3, DDPG ve PPO algoritmaları tercih edilmiştir. TORCS simülasyon ortamında gerçekleştirilen deneyler, DRL modellerinin eğitimi ve performans değerlendirmesi ile SHAP analizleri kapsamlı bir şekilde sunulmuştur. Eğitim ve test aşamalarında elde edilen bulgular TD3'ün en başarılı model olduğunu ortaya koymuştur. Modellerin karar verme süreçlerini etkileyen özellikler, SHAP teknikleri kullanılarak global ve lokal açıklanabilirlik analizleri ile değerlendirilmiştir. SHAP teknikleri ile global ve lokal açıklanabilirlik analizlerinde modellerin karar verirken etkilendikleri özellikler değerlendirilmiştir. TD3 modeline ait lokal açıklanabilirlik analizlerinde, TD3 modelinin uzman insan sürücü performansında eylemler sergilediği gözlemlenmiştir. Elde edilen bulgular, otonom araçların güvenilirliği ve hesap verebilirliği konusunda önemli katkılar sağlamaktadır. Bu tez, otonom sürüş sistemlerinin hem teknik hem de yasal gereksinimlere uygunluğunu sağlamada XAI yaklaşımlarının kritik bir rol oynadığını göstermektedir.

ANAHTAR KELİMELER: Açıklanabilir Yapay Zeka, Otonom Sürüş, Derin Pekiştirmeli Öğrenme, DDPG, PPO, TD3, TORCS

ABSTRACT

EXPLAINABLE ARTIFICIAL INTELLIGENCE (XAI) FOR DEEP REINFORCEMENT LEARNING BASED AUTONOMOUS DRIVING
MSC THESIS
MUHSİN KOMPAS
PAMUKKALE UNIVERSITY INSTITUTE OF SCIENCE
COMPUTER ENGINEERING
(SUPERVISOR: ASST. PROF. DR. İBRAHİM KÖK)

DENİZLİ, AUGUST 2024

In recent years, there has been a notable surge in the development of autonomous vehicle technology, driven by the rapid advancements in artificial intelligence and computer technologies. At the current time, several key industry stakeholders are engaged in active work on the development of autonomous vehicles. While the accelerated advancement of autonomous driving technology through data-driven and deep learning-based techniques has the potential to mitigate traffic issues and enhance efficiency, the intricacy and black box structure of these models present a considerable obstacle to the comprehensibility and transparency of decision-making processes. In this context, a number of legal, ethical and socio-psychological issues emerge. The application of explainable artificial intelligence (XAI) methodologies has emerged as a promising field of study with the potential to enhance the reliability, accountability and transparency of autonomous driving systems. In this context, XAI has the potential to play an important role in ensuring compliance with legal regulations, increasing user confidence and identifying deficiencies in software development. This thesis analyses and evaluates the performance of deep reinforcement learning (DRL) based autonomous driving systems and XAI approaches under different scenarios. Furthermore, the necessity for XAI methods to guarantee that autonomous vehicles comply with legal regulations and are accepted by society is emphasised. The TD3, DDPG and PPO methods were selected as the DRL models. The experiments conducted in the TORCS simulation environment are presented in a comprehensive manner, including the training and performance evaluation of DRL models and SHAP analyses. The findings obtained from the training and testing phases indicated that TD3 was the most successful model. The impact of specific features on the decision-making processes of the models was evaluated through global and local explainability analyses utilising SHAP techniques. In the local explainability analyses of the TD3 model, it was observed that the TD3 model exhibited actions that were comparable to those of expert human drivers. The obtained findings contribute to the advancement of the reliability and accountability of autonomous vehicles. This thesis demonstrates that XAI approaches play a pivotal role in ensuring that autonomous driving systems comply with both technical and legal requirements.

KEYWORDS: Explainable Artificial Intelligence, Autonomous Driving, Deep Reinforcement Learning, Artificial Intelligence, Deep Learning, XAI, AD, DRL, DDPG, PPO, TD3, TORCS

İÇİNDEKİLER

Sayfa

ŞEKİL LİSTESİ

Sayfa

TABLO LİSTESİ

Sayfa

SEMBOL LİSTESİ

$\mathbf{A_t}$	**:** t anında sisteme uygulanan eylem (Action).
$\boldsymbol{a^*}$	**:** Optimal eylem
distFromStart:	Start çizgisinden itibaren aracın aldığı mesafe.
distRaced	**:** Yarış başlangıcından itibaren aracın kat ettiği mesafe.
dist_(x)	**:** x açı konumundaki sensöre ait uzaklık bilgisi.
lastLapTime	**:** Son tamamlanan tur için süre.
$\mathbf{R_t}$	**:** Modelin eylemi sonucu t anında aldığı ödül.
$\mathbf{S_t}$	**:** t anında ortamdan gelen durum (State) bilgileri.
speedX	**:** Aracın hızı (km/h).
speedY	**:** Aracın yanal hızı.
speedZ	**:** Aracın dikey hızı.
wsv_x	**:** İlgili tekere ait dönüş hızı (rad/s).
$\boldsymbol{\gamma}$	**:** İndirgeme faktörü.
$\boldsymbol{\pi}$	**:** Politika
$\boldsymbol{\pi^*}$	**:** Optimal politika
$\boldsymbol{Q^{\pi}(s,a)}$	**:** Durum-eylem değer fonksiyonu (state-action value function)
$\boldsymbol{\pi(s)}$	**:** s durumundaki politika
$\boldsymbol{\pi(s\|\theta)}$	**:** Politika fonksiyonu, s durumunda alınacak a eylemini belirler.
$\boldsymbol{\phi}$	**:** Kritik ağının (critic network) parametreleri.
$\boldsymbol{\theta}$	**:** Aktör ağının (actor network) parametreleri.
$\boldsymbol{V^{\pi}}$	**:** Durum-değer fonksiyonu (state value function)
$\boldsymbol{y_t}$	**:** Gelecekteki ödülleri içeren hedef fonksiyonu
$\boldsymbol{L}$	**:** Kayıp fonksiyonu (Loss)
$\boldsymbol{\mathcal{D}}$	**:** Tekrar tamponu
$\boldsymbol{\epsilon}$	**:** Yumuşak güncelleme mekanizmasında küçük sabit, PPO'da ise klip oranı.

KISALTMA LİSTESİ

A3C	**:**	Asenkron Avantajlı Aktör-Kritik (Asynchronous Advantage Actor-Critic)
CNN	**:**	Konvolüsyonel Sinir Ağı (Convolutional Neural Network)
DDPG	**:**	Derin Deterministik Politika Gradyanı (Deep Deterministic Policy Gradient)
DNN	**:**	Derin Sinir Ağı (Deep Neural Network)
DPG	**:**	Deterministik Politika Gradyanı (Deterministic Policy Gradient)
DQN	**:**	Derin Q Ağı (Deep Q Network)
DRL	**:**	Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning)
GDPR	**:**	Genel Veri Koruma Yönetmeliği (General Data Protection Regulation)
LSTM	**:**	Uzun Kısa Süreli Bellek (Long Short-Term Memory)
MDP	**:**	Markov Karar Süreci (Markov Decision Process)
MSE	**:**	Ortalama Kare Hatası (Mean Squared Error)
NTSB	**:**	Ulusal Taşımacılık Güvenlik Kurulu (National Transportation Safety Board)
PPO	**:**	Proksimal Politika Optimizasyonu (Proximal Policy Optimization)
ReLU	**:**	Rectified Linear Unit
RL	**:**	Pekiştirmeli Öğrenme (Reinforcement Learning)
RC-araba:		Uzaktan Kumandalı Araba (Remote-Controlled Car)
SAE	**:**	Otomotiv Mühendisleri Topluluğu (Society of Automotive Engineers)
SHAP	**:**	Shapley Eklemeli Açıklamalar (SHapley Additive exPlanations)
TD3	**:**	İkiz Gecikmeli Derin Deterministik Politika Gradyanı (Twin Delayed Deep Deterministic Policy Gradient)
TORCS	**:**	The Open Racing Car Simulator
TRPO	**:**	Güven Bölgesi Politika Optimizasyonu (Trust Region Policy Optimization)
UAV	**:**	İnsansız Hava Aracı (Unmanned Aerial Vehicle)
UDP	**:**	Kullanıcı Veri Bloğu İletişim Kuralları (User Datagram Protocol)
XAI	**:**	Açıklanabilir Yapay Zeka (EXplainable Artificial Intelligence)
YZ	**:**	Yapay zeka (Artificial Intelligence—AI)

ÖNSÖZ

Bu yüksek lisans tez çalışması, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda yüksek lisans öğrenimimin bir parçası olarak hazırlanmıştır. "DERİN PEKİŞTİRMELİ ÖĞRENMEYE DAYALI OTONOM SÜRÜŞ İÇİN AÇIKLANABİLİR YAPAY ZEKA " başlığı altında gerçekleştirilmiş olan bu tez, otonom sürüş teknolojilerinde yapay zekâ algoritmalarının şeffaflığını ve anlaşılabilirliğini artırma amacını taşımaktadır. Bu çalışma, Pamukkale Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü'nde lisans eğitiminde gerçekleştirdiğim "OTONOM ARAÇLARDA GERÇEK-ZAMANLI ŞERİT ALGILAMA VE TAKİP SİSTEMİNİN GÖRÜNTÜ İŞLEM TABANLI OLARAK GERÇEKLEŞTİRİLMESİ" başlıklı bitirme tezimin doğal bir uzantısı niteliğindedir.

Bu süreçte değerli katkıları ve rehberliği için tez danışmanım Dr. Öğr. Üyesi İbrahim KÖK'e teşekkürü bir borç bilirim. Onun yönlendirmeleri ve bilimsel yaklaşımı, çalışmamın her aşamasında yol gösterici olmuştur. Ayrıca simülasyon çalışmalarım için kendi bilgisayarında çalışma imkanı sunduğu için kendisine özel bir teşekkür borçluyum.

Bana eğitimim boyunca destek sağlayarak motivasyonumun yüksek kalmasına yardımcı olan Prof. Dr. Ali Ersin ZÜMRÜTBAŞ'a sonsuz teşekkürlerimi sunarım. Tezimin teknik kısımlarında, özellikle araç dinamikleri konularında bana yol gösteren ve değerli bilgilerini paylaşan arkadaşım Ahmet Can SAK'a da teşekkür ederim.

Lisans eğitimimde çalıştığım ATAY Otonom Takımı'ndaki takım arkadaşlarıma ve danışman hocalarıma da bu vesileyle teşekkür etmek isterim. Otonom araçlar konusundaki bu ilk tecrübem, bugünkü çalışmalarımın temellerini atmamda büyük rol oynadı.

Bu süreçte yanımda olan annem, babam ve kardeşlerime manevi destekleri için teşekkür ederim.

1. GİRİŞ

Son yıllarda otonom araçların popülerliği dünya çapında giderek artmış ve çeşitli teknolojilerin önünü açmıştır (Baruch 2016). Otonom sürüş teknolojisinin yaygınlaşması trafikte yaşanan çeşitli sorunları azaltabilir ve iyileştirebilir (Nunes ve Axhausen 2021). 1984 yılında ABD Savunma İleri Araştırma Projeleri Ajansı (DARPA), Otonom Kara Aracı (ALV) Programı'nı başlatmak için ordu ile ortaklık kurmuştur. 1980'li yıllardan bu yana Carnegie Mellon Üniversitesi, Stanford Üniversitesi ve Massachusetts Teknoloji Enstitüsü gibi birçok üniversite, otonom sürüş araştırmalarına art arda katılmıştır. DARPA Grand Challenge (2004), otonom sürüş teknolojilerinin gelişiminde önemli bir rol oynamış bir yarışmadır. İlk kez 2004 yılında düzenlenen bu yarışma, tamamen otonom kara araçlarının zorlu bir parkuru insan müdahalesi olmadan tamamlamalarını hedeflemiştir. Otonom araç teknolojilerinin sınırlarını zorlayarak, sektörün hızlı bir şekilde ilerlemesini ve inovasyonların teşvik edilmesini sağlamıştır. DARPA Grand Challenge sayesinde, otonom sürüş sistemlerinin geliştirilmesi hız kazanmış ve günümüzde bu teknolojilerin yaygınlaşması için önemli bir temel oluşturulmuştur (Bacha ve diğ. 2004; Behringer ve diğ. 2004; Ozguner ve diğ. 2004). Apple, Audi, BMW, Ford, General Motors, Google, Honda, Mercedes, Nissan, Nvidia, Tesla, Toyota ve Volkswagen gibi markalar 2010'lu yılların başından beri insansız araçlar alanında faaliyet göstermeye başlamıştır (Greenblatt 2016). DARPA Grand Challenge'den sonra, veri tabanlı ve makine öğrenmesi temelli teknikler otonom sürüşte yaygın olarak kullanılmaya başlanmıştır. Son yıllarda ise derin öğrenme temelli yöntemler giderek artmıştır. Abu Dabi Otonom Yarış Ligi (A2RL 2024), ASPIRE tarafından 2024 yılında düzenlenen ve Yas Marina Pisti'nde otonom araçların yarıştığı ilk etkinlik olmuştur. Bunun yanı sıra SAE International (Society of Automotive Engineers), otonom sürüş teknolojileri alanında SAE J3016 (2021) standardını tanımlayarak bu teknolojilerin sınıflandırılmasını ve anlaşılmasını kolaylaştırmıştır. Bu standart, endüstri genelinde yaygın olarak kabul edilmiş ve benimsenmiştir; böylece otonom araçların geliştirilmesi ve değerlendirilmesi için ortak bir çerçeve sunulmuştur. Sürüş otomasyon seviyelerini 0 ila 5 arasında sınıflandıran bu standartta, seviye yükseldikçe otomasyon derecesi artarken; sistemin karmaşıklığı ve gerektirdiği uzmanlık da

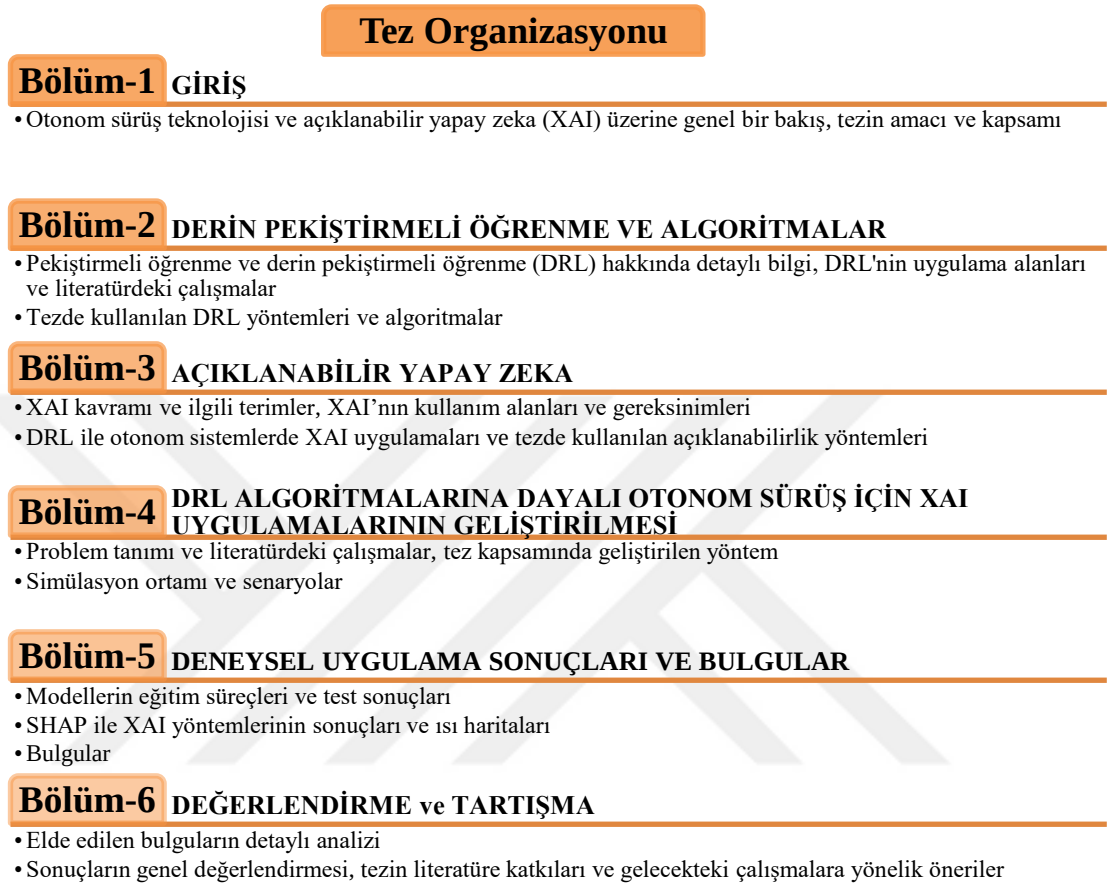
artmaktadır. Bu sistemlerin geliştirilmesi maliyeti ve belirsizliği de arttırmaktadır. Derin öğrenme modellerinin yapısı bu tür karmaşık görevleri yerine getirmede yüksek başarımlar göstermiştir. Ancak modellerin iç işleyişini karmaşıklığı ve kara-kutu yapıları nedeniyle nasıl çalıştığını anlamak zor olmakta ve belirli bir kararı neden verdiğini açıklamayı zorlaştırmaktadır (LeCun ve diğ. 2015).

Otonom sürüş teknolojisinin sunduğu potansiyel faydalar, beraberinde çeşitli teknik, yasal ve etik zorlukları da getirmektedir. Bu zorlukların başında, otonom araçların karar verme süreçlerinin anlaşılabilir ve açıklanabilir olmaması gelmektedir. Derin öğrenme modellerinin karmaşıklığı, bu süreçlerin genellikle bir kara kutu olarak algılanmasına neden olmakta ve güvenilirlik, hesap verebilirlik gibi konularda soru işaretleri doğurmaktadır (Doshi-Velez ve Kim 2017). Otonom araçların güvenilirliğini artırmak için açıklanabilir yapay zeka (Explainable Artificial Intelligence—XAI) yaklaşımları büyük önem taşımaktadır. XAI, yapay zeka sistemlerinin kararlarını anlaşılır ve takip edilebilir bir şekilde açıklayarak, kullanıcıların ve geliştiricilerin bu sistemlere olan güvenini artırmayı hedeflemektedir. Bu tez, derin pekiştirmeli öğrenme (Deep Reinforcement Learning—DRL) algoritmalarına dayalı otonom sürüş sistemlerinin açıklanabilirliğini incelemekte ve bu sistemlerin çeşitli senaryolar altında nasıl performans gösterdiğini değerlendirmektedir. Ayrıca otonom araçların yasal düzenlemelerle uyumlu hale getirilmesi ve toplum tarafından kabul edilmesini sağlamak amacıyla gerekli olan güvenin oluşturulabilmesi için hesap verebilirlik ve şeffaflık konularına da odaklanmaktadır. Özellikle GDPR (2018) gibi düzenlemeler, otonom araçların topladığı verilerin ve bu verilere dayanarak alınan kararların şeffaf olmasını zorunlu kılmaktadır (Goodman ve Flaxman 2017). Bu yüzden açıklanabilir yapay zeka yaklaşımları, otonom sürüş sistemlerinin hem teknik hem de yasal gereksinimlere uygunluğunu sağlamada kritik bir rol oynamaktadır. Bu bağlamda bu tezde DRL tabanlı otonom sürüşün açıklanabilirliği konusuna odaklanılmıştır.

Açıklanan problemler ve bu problemlere yaklaşımlardan hareketle tezin organizasyonu Şekil 1.1'de verilmiş olup bölümlerin içerikleri aşağıda özetlenmiştir.

Bölüm 2'de, pekiştirmeli öğrenme, DRL ve tezde kullanılan DRL algoritmaları hakkında detaylı bilgi sunulmaktadır. Bu bölümde öncelikle MDP, Bellman Denklemlerinden ve pekiştirmeli öğrenme kavramlarından bahsedilmiştir. Ardından DRL hakkında bilgi verilmiş, uygulama alanları ve literatürde yapılan çalışmalara

değinilmiştir. Daha sonra bu tez kapsamında kullanılan DRL yöntemleri ve algoritmaları açıklanmıştır.



Şekil 1.1: Tezin organizasyonu.

Bölüm 3'te, açıklanabilir yapay zeka (XAI) hakkında detaylı bilgi sunulmaktadır. Öncelikle XAI kavramı açıklanmış, XAI ile ilgili terimler hakkında tanımlamalar yapılmıştır. Ardından XAI'nın kullanım alanlarından ve gereksiniminden bahsedilmiştir. Daha sonra DRL ve otonom sistemler için XAI'dan bahsedilmiştir. Son olarak tezde kullanılan açıklanabilirlik yöntemlerine değinilmiştir.

Bölüm 4'te, öncelikle problem tanımı, DRL ve XAI hakkında literatürde yer alan çalışmalar sunulmaktadır. Ardından tez kapsamında geliştirilen yöntemden bahsedilmektedir. Bu bölümde son olarak simülasyon ortamı hakkında detaylı bilgi verilerek, eğitim ve test koşullarına ait senaryolar ayrıntılı olarak işlenmektedir.

Bölüm 5'te, modellerin eğitimi ve test sonuçlarına ilişkin bilgiler verilmekte ve sonuçlar kapsamlı olarak sunulmaktadır. Ayrıca SHAP ile XAI yöntemlerine ait sonuçlar sunularak ısı haritaları ile desteklenmektedir.

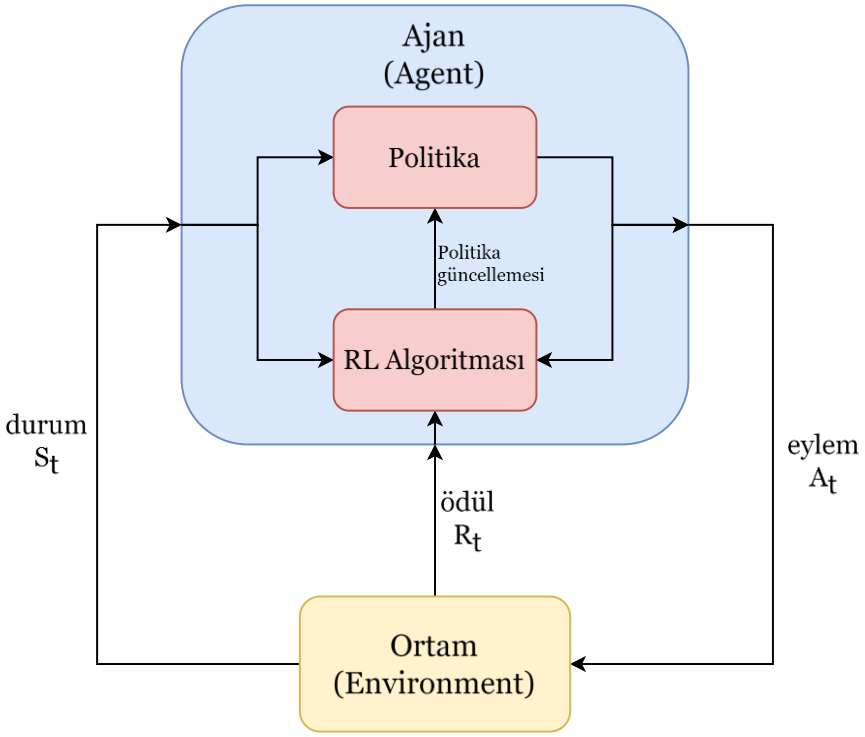
Bölüm 6'da, tez çalışması kapsamında elde edilen bulgular ortaya koyularak sonuçlar değerlendirilmiştir. Buna ek olarak tezin literatüre katkılardan bahsedilerek gelecekteki çalışmalara yönelik öneriler sunulmuştur.

2. DERİN PEKİŞTİRMELİ ÖĞRENME VE ALGORİTMALAR

Bu bölümde tez kapsamında geliştirilen çalışmaların temellerinden birini oluşturan derin pekiştirmeli öğrenme hakkında detaylı bilgi sunulmaktadır.

2.1 Pekiştirmeli Öğrenme (Reinforcement Learning—RL) ve Markov Karar Süreçleri

Pekiştirmeli öğrenmede ajanın, çevre ile etkileşime girmesi beklenir. Her bir adımda ortamın bir durumu vardır. Bu durumlar ajanın konumu, hızı, ortamdaki sensörlerden alınan bilgiler olabilir. Ajanın her adımda mevcut durum bilgilerinin gözlemini (observation) elde etmesi gerekir. Ardından ajan gözlem ve politikasına göre bir eylemde bulunur. Eylem ortama uygulanır ve buna bağlı olarak da ajanın ortamdaki durumu değişir. Daha sonra ortamdan yeni bir gözlem ve ödül alır. Ödül pozitif veya negatif olabilir. Negatif olması durumunda ceza olarak da belirtilebilir. Ajan gözleme göre tekrar bir eylemde bulunur. Bu duruma örnek **Şekil 2.1**'de verilmiştir.



Şekil 2.1: Pekiştirmeli öğrenmenin diyagramı.

Pekiştirmeli öğrenme, bir ajan (agent) ve ortam (environment) arasında etkileşim yoluyla öğrenmesi sürecini Markov Karar Süreçleri (Markov Decision Process—MDP) olasılık formunda ifade eder (Plaat 2022). MDP, bir ajan ve çevre arasındaki etkileşimlerin modellendiği bir sistemdir ve dört ana bileşenden oluşur: durumlar (states), eylemler (actions), geçiş olasılıkları (transition probabilities) ve ödül fonksiyonu (reward function).

S; durum uzayı kümesi olup, ortamın mevcut durumunu ve ajanın eylemlerinin sonucunda oluşan sistemin tüm durumlarını ifade eder. A, eylem uzayıdır. Ajanın eylem uzayında alabileceği tüm eylemleri içerir. $P(s'|s,a)$, ajanın bir eylem gerçekleştirdiğinde bir durumdan diğerine geçiş olasılığını ifade tanımlar. Burada a, s durumunda yapılmış eylemi ve s' eylem sonucundaki durumu ifade eder. $R(s,a)$, ödül fonksiyonunu temsil eder. Ajan s durumunda a eylemini gerçekleştirdiğinde elde ettiği ödülü belirtir.

MDP'nin temel özelliklerinden biri Markov özelliğidir. Bu özellik sistemin gelecekteki durumunun sadece mevcut duruma ve gerçekleştirilen eyleme bağlı olduğunu, geçmişe bağımlı olmadığını ifade eder. MDP'de ajanın amacı, en yüksek toplam ödülü elde edecek şekilde bir eylem stratejisi (politika, π) geliştirmektir. Politika her durumda hangi eylemin seçileceğini belirler. Optimal politika (π^*) ajanı ulaşılabilen en yüksek ödüle ulaştıran politikadır. Ajanın eylem seçimi, $s \in S$ durumunu $\pi(A|s)$ eylemi üzerinde bir olasılık dağılımına eşleyen politika tarafından belirlenir. RL ajanın gelecekteki alacağı ödülleri maksimize etmeyi hedefler. $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ gelecekteki alabileceği ödülleri belirtir; burada γ indirim faktörü (discount factor) 0 ile 1 arasında değer alır ve gelecekteki ödüllerin mevcut durumdaki değerine indirgenmesini tanımlar. Ancak bir eylemi gerçekleştirirken gelecekteki durumlar genellikle tam olarak hesaplanamaz. Bu nedenle olası tüm durumları göz önünde bulundurularak ve durum geçiş olasılığına dayanarak uzun vadeli getiriler hesaplanır.

Değer fonksiyonu beklenen uzun vadeli getirileri temsil eder. Değer fonksiyonu, durum değer fonksiyonu ve durum-eylem değer fonksiyonu olarak iki şekilde hesaplanır. Durum değer fonksiyonu (state value function) mevcut durum için uzun vadeli beklenen getirilerdir ve $V^{\pi}(s)$ ile eşitlik (2.1)'deki gibi temsil edilir;

burada π politikayı belirtir. Benzer şekilde π politikası altında durum-eylem değer fonksiyonu (state-action value function) a eylemini s durumundan itibaren beklenen getiriler için $Q^{\pi}(s,a)$ ile hesaplanır. Beklenen değer $\mathbb{E}[\cdot]$ ajanın π politikasını izlemesi durumunda rastgele bir değişkenin beklenen değerini gösterir.

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t|S_t = s] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle| S_t = s\right] \tag{2.1}$$

Benzer şekilde π politikası için eylem-değer fonksiyonu Q_{π} , denklem (2.2)'deki gibi ifade edilir. Burada a eyleminin s durumunda π politikası altında $Q_{\pi}(s,a)$ ile tanımlanır.

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}[G_t|S_t = s, A_t = a] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle| S_t = s, A_t = a\right] \tag{2.2}$$

Eşitlik (2.1) ve (2.2)'nin her ikisi de bir durumun değeri ile kendisinden sonra gelen durumların değerleri arasındaki rekürsiv ilişkileri sağlar. Bu özellik Bellman Denklemlerinden türeterek eşitlik (2.3)'teki gibi gösterilebilir. Burada $s_{t+1}\sim E$ bir sonraki durumun E ortamından (environment) örneklendiği anlamına gelir ve $a_{t+1}\sim\pi$, π politikasının bir sonraki eylemi belirlediğini gösterir.

$$\begin{aligned} V^{\pi}(s_t) &= \mathbb{E}_{a_t\sim\pi, s_{t+1}\sim E}[r(s_t,a_t) + \gamma V^{\pi}(s_{t+1})] \\ &= \sum_{a\in A}\pi(a|s)\sum_{s'\in S, r\in R}P(s',r|s,a)[r + \gamma V^{\pi}(s')] \\ Q^{\pi}(s_t,a_t) &= \mathbb{E}_{s_{t+1}\sim E}\Big[r(s_t,a_t) + \gamma\mathbb{E}_{a_{t+1}\sim\pi}\left[Q^{\pi}(s_{t+1},a_{t+1})\right]\Big] \\ &= \sum_{a\in A}\pi(a|s)\sum_{a'\in S, r\in R}P(s',r|s,a)\left[r + \gamma Q^{\pi}(s',a')\right] \end{aligned} \tag{2.3}$$

$r(s_t,a_t)$ başlangıç durumu ve gerçekleştirilen eylem için ödül değeridir. Amaç en uygun politika olan π^*'ı bulmaktdır. Optimal politikayı bulmak için eşitlik (2.4) ve

(2.5) bulunmalıdır. Bunlar sırasıyla optimal değer fonksiyonu ve optimal eylem-değer fonksiyonudur. Bu Bellman optimalite denklemleriyle eşitlik (2.6)'daki gibi bulunur.

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{2.4}$$

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{2.5}$$

$$\begin{aligned} V^*(s_t) &= \max_{a} \mathbb{E}_{s_{t+1}\sim E}[r(s_t,a) + \gamma V^*(s_{t+1})] \\ &= \max_{a} \sum_{s'\in S, r\in R} P(s',r|s,a)[r + \gamma V^*(s')] \\ Q^*(s_t,a_t) &= \mathbb{E}_{s_{t+1}\sim E}\left[r(s_t,a) + \gamma \max_{a'}[Q^*(s_{t+1},a')]\right] \\ &= \sum_{s'\in S, r\in R} P(s',r|s,a)\left[r + \gamma \max_{a'} Q^*(s',a')\right] \end{aligned} \tag{2.6}$$

2.2 Derin Pekiştirmeli Öğrenme (Deep Reinforcement Learning—DRL)

Derin öğrenme, çok katmanlı derin sinir ağlarını kullanarak karmaşık ve yüksek boyutlu ortamlar için yaklaşımlar bulmayı amaçlar. Bu yöntem, veriden fonksiyonları yaklaşık olarak hesaplamayı hedefleyen makine öğrenmesi algoritmalarının gelişmiş bir biçimidir. Geleneksel makine öğrenmesi algoritmaları, doğrusal regresyon, karar ağaçları, rastgele ormanlar, destek vektör makineleri ve yapay sinir ağları gibi yöntemler kullanarak veriler üzerinde tahmin modelleri öğrenir. Derin öğrenme, sinir ağlarının çok katmanlı yapısını kullanarak daha karmaşık ve yüksek boyutlu problemleri çözebilme kapasitesine sahiptir (LeCun ve diğ. 2015).

Pekiştirmeli öğrenme, bir ajan ile bir ortam arasında etkileşim kurarak öğrenmeyi amaçlar. Ajan, çeşitli eylemleri seçer ve bu eylemler sonucunda ortamdan geri bildirim alır. Bu geri bildirim, ödül sinyalleri şeklinde olur ve ajan, bu ödüllere

göre hangi eylemlerin en iyi sonuçları verdiğini öğrenir (Kaelbling ve diğ. 1996). Pekiştirmeli öğrenme, etkileşim ve deneme-yanılma yoluyla öğrenmeyi içerir; bu da ajanın kendi kendine veri oluşturmasını sağlar. Pekiştirmeli öğrenme, mevcut veri setlerine ihtiyaç duymaz, bunun yerine ajan, ortamdan aldığı geri bildirimlerle öğrenir.

Derin pekiştirmeli öğrenme (Deep Reinforcement learning—DRL) ise derin öğrenme yöntemlerini pekiştirmeli öğrenme ile birleştirerek yüksek boyutlu ve karmaşık problemleri çözmeyi amaçlar (Plaat 2022). Bu yöntem yüksek boyutlu, etkileşimli öğrenme süreçlerini mümkün kılar. DRL, derin öğrenme yöntemlerini kullanarak pekiştirmeli öğrenmenin öğrenme sürecini iyileştirir ve genişletir (X. Wang ve diğ. 2024). Sistem, doğru davranışları teşvik eden pozitif ödüller ve yanlış davranışları cezalandıran negatif ödüller ile eğitilir. Bu süreçte yapay sinir ağları kullanılarak alınan kararların karmaşıklığı artar ve modelin çeşitli senaryolara uyum sağlama kapasitesi geliştirilir.

Kontrol politikasını öğrenmek için DRL kullanan ilk uygulamalardan biri yüksek boyutlu sensör girdisini doğrudan alan Derin Q-Ağı (DQN) olmuştur (Mnih ve diğ. 2013). Model konvolüsyonel sinir ağı (CNN) ile temsil edilmiş ve muhtemelen tüm eylemleri tahmin etmiştir. Klasik Atari 2600 oyunları üzerindeki test sonuçlarına göre DQN ajanının önceki tüm algoritmalardan daha iyi performans göstererek bazı oyunlarda insan seviyesinde performansa ulaşabildiğini göstermiştir.

DQN, hedef değeri hesaplarken bir sonraki zaman adımının optimum değerini benimser. Bununla birlikte eylemi seçmek ve hedef değeri hesaplamak için aynı ağı kullanır; bu da değerin fazla tahmin edilmesine (overestimation) neden olabilmektedir. Google DeepMind, aşırı tahmin sorununa yönelik olarak optimum eylemi (action) seçmek için bir davranış ağı (behavior network) ve hedef değeri hesaplamak için bir hedef ağı (target network) kullanan Double DQN algoritmasını önermiştir (van Hasselt ve diğ. 2016). Bu algoritma daha doğru değer tahmini sağlayabilmiş ve Atari oyunlarında daha iyi bir performans göstermiştir.

DQN'nin bir başka geliştirmesi de iki ayrı tahmin sunan Dueling DQN'dir (Ziyu Wang ve diğ. 2016). Durum değeri (state value) fonksiyonu ve durum-eylem (state-action) avantaj fonksiyonu, görüntülerden özellikleri öğrenmek için ortak bir CNN kullanır. Bir çıktıyı iki çıktıya dönüştürür ve iki çıktının toplamı durum-eylem

değeridir. Bu düello mimarisi, her bir eylemin durum üzerindeki etkisini öğrenmeden durum değerini öğrenebilir. Durum-eylem değerinden ayrıştırılan her bir parçanın pratik bir anlamı vardır ve bu parçalardan birçok değerli bilgiler çıkarılabilir.

Google DeepMind, Rainbow (Hessel ve diğ. 2017) adlı birleştirilmiş bir model geliştirmek için DQN'nin altı uzantısını birleştirmiştir; bu modelde çift Q-öğrenme, düello ağı, farklı geçişlere farklı ağırlıklar vermek için öncelikli tekrar belleği (prioritized replay (Schaul ve diğ. 2016)), çok adımlı öğrenme, değer olasılığını histogram şeklinde temsil etmek için dağılımsal ağ (distributional network (Marc ve diğ. 2017)), verimli keşfe (efficient exploration) yardımcı olmak için ağırlıklara parametrik gürültü eklenmiş gürültülü ağ yer almıştır(Fortunato ve diğ. 2019). Deneyler sonucunda bu kombinasyonların Atari oyunlarında en gelişmiş performansı elde edebileceğini göstermiştir.

DQN ve varyantları yalnızca ayrık eylem uzayına (discrete action space) sahip problemleri çözmek için uygulanabilir ancak gerçek dünyadaki birçok karmaşık görev sürekli eylem uzayına (continuous action space) sahiptir. Bu nedenle sürekli kontrole uygulanabilir takviye öğrenme algoritması geliştirilmesine ihtiyaç duyulmuştur. Google DeepMind, stokastik politika gradyanı yerine eylem-değer fonksiyonunun beklenen gradyanını hesaplayan sürekli eylemlerle bir deterministik politika gradyan algoritması (DPG) önermiştir (Silver ve diğ. 2014).

Google DeepMind; DPG algoritmasına dayalı, aktör-eleştirel (actor-critic) bir yapı kullanan ve DQN'nin iki önemli iyileştirmesini, yeniden oynatma belleğini ve hedef ağını tanıtan derin bir deterministik politika gradyan algoritmasını (Deep Deterministic Policy Gradient—DDPG) önermiştir. DDPG, birçok fizik görevinde politikaları doğrudan ham piksellerden öğrenebilmektedir (Timothy ve diğ. 2019).

U.C. Berkeley'den Schulman ve diğ. (2017a), sinir ağı tarafından temsil edilen büyük doğrusal olmayan politikaları optimize etmek için etkili olan ve politikanın monotonik gelişimini garanti eden bir güven bölgesi politika optimizasyon (Trust Region Policy Optimization—TRPO) algoritması geliştirmiştir. Çok sayıdaki farklı görevler üzerinde sağlam bir performans göstermiştir. TRPO'nun hesaplanması karmaşık olduğundan, hesaplama hızı diğer politika gradyan algoritmalarına kıyasla yavaştır. Bu nedenle OpenAI, yalnızca TRPO'nun avantajlarına sahip olmakla

kalmayıp aynı zamanda uygulaması çok daha kolay olan bir proksimal politika optimizasyon algoritması (PPO) önermiştir. PPO, mini yığın (mini-batch) örneği başına birden fazla güncellemeye olanak tanır; bu nedenle daha iyi bir örnek verimliliğine sahiptir. Yapılan deneyler, PPO'nun performansının bir dizi benchmark görevinde diğer politika gradyan yöntemlerini geride bıraktığını göstermiştir (Schulman ve diğ. 2017b).

DDPG'deki yaygın hata Q-fonksiyonunun aşırı tahmin edilmesidir ve bu da Q-fonksiyonunun hatalarından faydalanarak (exploit) politikanın kırılmasına (breaking) yol açabilir. McGill Üniversitesi'nden Fujimoto ve diğ. (2018), bu sorunu üç kritik püf noktası sunarak çözebilecek İkiz Gecikmeli DDPG'yi (Twin Delayed DDPG—TD3) ortaya koymuştur. Bu üç püf noktası şunlardır:

- Bir yerine iki Q-fonksiyonu öğrenmek ve hedef değeri hesaplamak için daha küçük değeri kullanmak,
- aktörü eleştirmenden daha az sıklıkta güncellemek,
- hedef eyleme gürültü eklemek.

Böylece politika, eylemler üzerinde Q-fonksiyonunu yumuşatarak (smoothing) Q-fonksiyonu hatalarından yararlanma olasılığını azaltır.

Yukarıda DRL algoritmalarının bazılarının belirtilmesinden sonra DRL'nin farklı alanlardaki bazı uygulamalarına yer verilecektir. DRL ile elde edilen ilk büyük gelişme insan Go şampiyonunu 4-1 yenen AlphaGo olmuştur. Monte Carlo ağaç araması ile birleştirilmiş bir değer ağı ve bir politika ağı kullanmıştır (Silver ve diğ. 2016). Derin sinir ağları, insan deneyimlerinden denetimli öğrenme (supervised learning) ve kendi kendine oyundan takviye öğrenme yoluyla eğitilmiştir; bu nedenle AlphaGo, denetimli eğitim verileri olarak oyuncuların bilgisine ihtiyaç duymuştur. Ardından Google DeepMind, herhangi bir insan deneyimi ve rehberliği olmadan sadece oyun kuralları ile derin takviye öğrenimi ile eğitilen AlphaGo Zero'yu geliştirmiştir (Silver ve diğ. 2017). AlphaGo Zero, Monte Carlo ağaç arama yönteminin iyileştirilmesi olan eylemini ve oyunun galibini tahmin etmek için bir sinir ağı kullanmıştır. AlphaGo Zero, politikayı sıfırdan kendi kendine oynayarak öğrenmiş ve önceki AlphaGo'ya karşı 100'e 0'lık bir skor elde etmiştir.

DRL ile masa oyunları ve video oyunlarında büyük başarım elde edilmiştir öte yandan gelecek vadeden uygulama alanlarından biri robotik olmuştur. Manipülasyon, robot navigasyonu ve otonom sürüş dahil olmak üzere birçok robotik görevi başarıyla çözmek için çeşitli uygulamalarda kullanılmıştır.

Popov ve diğ. (2017), DDPG'yi iki uzantıyla ortam adımı başına çoklu mini yığın güncellemeleriyle ve veri toplama ve ağ eğitimini birden fazla robota tahsis etmek için dağıtılmış versiyonla, bir nesneyi kavramak ve ardından diğer nesnenin üzerine tam olarak istiflemek için hünerli bir manipülatörü başarıyla eğitmek için kullanmıştır. OpenAI (2019), görüş tabanlı nesne yönlendirmeyi öğrenmek üzere fiziksel bir hünerli eli eğitmek için asimetrik bir Aktör-Eleştirmen yaklaşımıyla Proksimal Politika Optimizasyonunu (PPO) benimsemiştir. Politika, tamamen simülasyonda eğitilmiştir fakat alan rastgeleleştirmesi yoluyla gerçek dünya robotuna aktarılabilmiştir. Bu yöntem herhangi bir insan gösterimine bağlı değildir ancak manipülasyondaki bazı insan davranışlarını doğal olarak sunabilmektedir.

Tai ve diğ. (2017) mobil robot ile haritasız navigasyon görevinde asenkron DDPG yöntemini uygulamıştır. Herhangi bir manuel özellik çıkarımı ve ön gösterim olmadan sadece seyrek aralık bulgularını girdi olarak alarak uçtan uca bir yöntemle eğitilmiştir. Eğitilen ajan, görülmeyen simüle ve gerçek ortamlara da başarıyla aktarılabilmiştir. Yuke Zhu ve diğ. (2016) DRL'yi iç mekânda mobil robot görsel navigasyonuna uygulamıştır. Bu çalışmada yüksek kaliteli 3B ortam sağlayabilen bir çerçeve önermişlerdir. Uçtan uca eğitilen bu model ince ayar (fine-tuning) ile gerçek hayata uygulanabilmektedir. Fan ve diğ. (2018) çoklu robot sistemi için navigasyon ve kazadan kaçış politikası izleyen bir çoklu robot sistemi önermiştir. Politika PPO ile karmaşık ortamlarda, çok senaryolu ve çok aşamalı eğitim çerçevesi ile eğitilmiştir. Politikanın sağlamlığını ve etkinliğini arttırmak için hibrit bir kontrol stratejisi izlenmiştir. Sanal olarak üretilmiş ve modellerin önceden görmediği haritalarda da test edilip doğrulanmıştır. Xi Chen ve diğ. (2018), ajanın karmaşık ortamlarda gezinmeyi öğrenmesi için A3C kullanılan uçtan uca bir DRL çerçevesi önermiştir. Kamera görüntüleri ve derinlik bilgileri ile mevcut konuma daha önce gelinip gelinmediğinin analizi yapılmış ve sonlandırma yaparak performansın arttırılması önerilmiştir. Bu yöntem, karmaşık 3B labirentte ham görüntü girdisinden navigasyon becerilerini öğrenebilmiş ve insan düzeyinde performansa yaklaşmıştır.

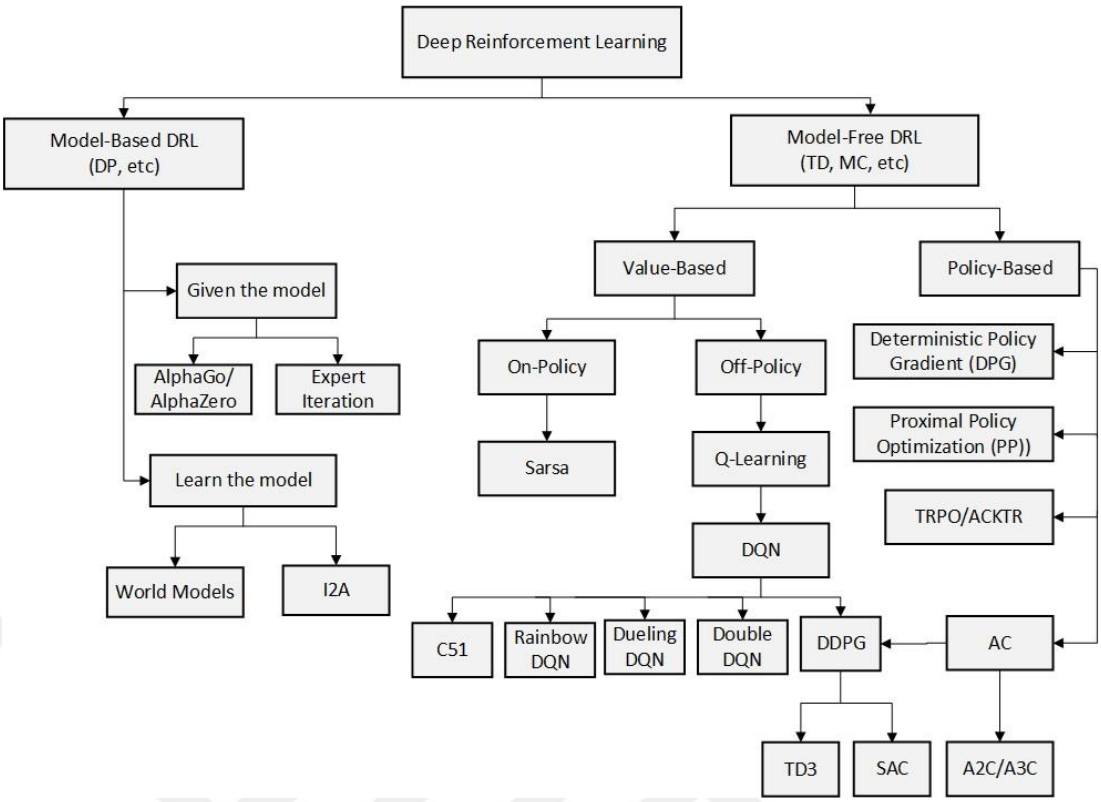
Otonom araçlar son yıllarda oldukça popüler bir araştırma ve çalışma alanıdır (Baruch 2016). Derin pekiştirmeli öğrenme, sürüş politikasını öğrenmek için oldukça kullanışlı bir teknik olarak kabul görmektedir; literatürde bu konuyla ilgili yapılan çalışmalar şunlardır. Sallab ve diğ. (2017) sürüş davranışlarını öğrenmek için ham sensör verilerini girdi olarak kullanan ve sürüş eylemlerini çıktı olarak veren, DQN aracılığıyla uçtan uca bir yöntem kullanmıştır. Kısmi gözlemlenebilir senaryolarla başa çıkmak için LSTM kullanmışlardır. Ayrıca hesaplamayı azaltmak için sensör verilerinden ilgili bilgileri çıkarmak için dikkat (attention) modelini entegre etmişlerdir. Bu çerçeve, araba yarışı simülatörü TORCS üzerinde test edilmiştir. Simülasyon sonuçları sürüş davranışlarının başarılı bir şekilde öğrendiğini göstermiştir. Sistem araba yarışı simülatörü TORCS üzerinde test edilmiştir. Simülasyon sonuçları sürüş davranışlarını başarılı bir şekilde öğrenildiğini göstermiştir. Pan ve diğ. (2017) otonom sürüş için A3C tabanlı sanaldan gerçeğe (virtual-to-real) bir DRL çerçevesi önermiştir. Simülasyondaki sanal görüntüleri ortak sahne segmentasyonlarına dayalı olarak yaklaşık gerçekçi görüntülere dönüştürmek için gerçekçi bir çeviri ağı kullanmışlardır. Böylece bu sentezlenmiş gerçekçi görüntülerle eğitilen takviyeli öğrenme ajanı, gerçek dünyadaki sürüş ortamlarına uyum sağlayabilmiştir. Jaritz ve diğ. (2018) A3C'yi sadece kameradan alınan RGB görüntüsünü girdi olarak alıp, sürüş politikasını uçtan uca eğitmek için kullanmıştır. Hızlı yakınsama, sağlam performans sağlayan yeni bir ödül fonksiyonu ve eğitim stratejisi önermişlerdir. Genelleme yeteneği olarak daha önceden görülmeyen ortamlarda da doğrulanmıştır. Sharifzadeh ve diğ. (2017) otonom sürüş senaryosunda ödülleri elde etmek için DQN tabanlı bir ters takviye öğrenimi benimsemiştir. Yeterli eğitimden sonra ajanların insansı sürüş performansına yakın, çarpışmasız hareketler ve şerit değiştirme davranışları sergilediği gözlemlenmiştir. Kaushik ve diğ. (2018) sollama davranışlarını bir otonom araç modeline öğretmek için DDPG kullanmıştır. Model, müfredat öğrenmesine (curriculum learning) benzer bir yöntemle eğitilmiştir. Burada model ilk önce yolda gitmeyi öğrenmiş ardından diğer araçları sollamayı öğrenmiştir. Bu yöntem modelin araç ile sollama manevralarını başarıyla yapmasını sağlamıştır. Yilun Chen(Yilun Chen 2018), otonom aracın şerit değiştirme davranışlarını öğrenmesi için modeli eğitirken hiyerarşik bir eylem alanına sahip Derin Tekrarlayan Deterministik Politika Gradyanı (Deep Recurrent Deterministic Policy Gradient) yöntemini kullanmıştır. Ayrıca bir görüntünün en önemli bölgesini tespit eden uzamsal dikkat ve son birkaç kareye farklı ağırlıklar atayan zamansal dikkat de

dahil olmak üzere dikkat mekanizmasını tanıtmıştır. Böylece araç daha iyi performans sergileyerek şerit değiştirme davranışları gerçekleştirmiştir.

DRL benzer kontrol alanlarında da kullanılmaktadır. Yu ve diğ. (2017) otonom su altı aracı için yörünge izleme problemini çözmek üzere DDPG algoritmasını uygulamıştır. H. Wu ve diğ. (2019), otonom su altı aracının derinlik kontrol problemini çözmek için DDPG'yi kullanmıştır. Ayrıca kontrol performansını artırmak için öncelikli deneyim tekrarı sunmuşlardır. Simülasyon sonuçları kullanılan yöntemin, LQR ve MPC gibi model tabanlı kontrolörlerden daha iyi performans elde etmişlerdir. Ayrıca insansız hava aracı (UAV) ile çeşitli görevlerde kullanılmıştır. Bu görevleri gerçekleştirmek için PPO, DDPG gibi DRL yöntemlerinden yararlanıldığı belirtilmiştir (Bohn ve diğ. ; Bouhamed ve diğ. 2020; Koch ve diğ. 2019; Jie Xu ve diğ. 2019).

Şekil 2.2'de DRL yöntemlerinin model tabanlı (model-based) ve model bağımsız (model-free) olarak iki ana kategoriye ayrıldığı gösterilmektedir. Model tabanlı yöntemler, ortamın modelini öğrenmeyi veya kullanmayı hedeflerken, model bağımsız yöntemler doğrudan ödül sinyalleri ile öğrenir. Model bağımsız yöntemler ise kendi içinde değer tabanlı (value-based) ve politika tabanlı (policy-based) yöntemlere ayrılır.

Değer tabanlı yöntemler durum-değer fonksiyonlarını öğrenirken, politika tabanlı yöntemler doğrudan politika fonksiyonlarını öğrenir. Ayrıca aktör-eleştirmen yöntemleri de hem değer tabanlı hem de politika tabanlı yaklaşımların avantajlarını birleştirir. Bu kategoriler, DRL yöntemlerinin çeşitli uygulama alanlarındaki performansını ve uygunluğunu belirlemekte önemli bir rol oynar.



Şekil 2.2: Derin pekiştirmeli öğrenme algoritmalarının taksonomisi (Azar ve diğ. 2021).

DRL, temel olarak üç ana bileşene dayanır: değer tabanlı yöntemler (value-based methods), politika tabanlı yöntemler (policy-based methods) ve aktör-eleştirmen (actor-critic) yöntemler. Değer tabanlı yöntemler, her durum-eylem çiftinin değerini öğrenmeye çalışır. Bu yöntemlerde Q-değer fonksiyonu (Q-value function) kullanılarak, her durum-eylem çiftinin değeri hesaplanır. Örneğin DQN yöntemi, Q-değer fonksiyonunu derin sinir ağları ile yakınsayarak hesaplar ve yüksek boyutlu durum uzaylarında etkin öğrenme sağlar.

Politika tabanlı yöntemler, doğrudan optimal politikayı öğrenir ve bu politikayı optimize eder. Bu yöntemlerde, ajan eylem seçme olasılıklarını doğrudan öğrenir. Örneğin PPO, politikaları güvenli ve etkin bir şekilde güncelleyerek, sürekli kontrol problemlerinde başarılı sonuçlar elde eder.

Aktör-eleştirmen yöntemler ise hem değer tabanlı hem de politika tabanlı yaklaşımların avantajlarını birleştirir. Aktör (actor) eylemleri seçerken, eleştirmen (critic) bu eylemlerin değerini değerlendirir ve geri bildirim sağlar. Böylece daha stabil

ve hızlı öğrenme süreçleri elde edilir. Örneğin DDPG ve TD3 gibi algoritmalar, aktör-eleştirmen yapısı ile sürekli kontrol problemlerinde etkin çözümler sunar.

2.3 Kullanılan DRL Algoritmaları

Bu çalışmada otonom sürüş sistemlerinin geliştirilmesi için üç farklı DRL algoritması kullanılmıştır. Bu yöntemler sürekli eylem uzaylarında çalışmaya uygundur. İncelenen yöntemler arasında Derin Deterministik Politika Gradyanı, İkiz Gecikmeli DDPG ve Proksimal Politika Optimizasyonu bulunmaktadır. Bu algoritmaların ayrıntıları aşağıda sunulmuştur.

2.3.1 Derin Deterministik Politika Gradyanı (Deep Deterministic Policy Gradient—DDPG)

Derin Deterministik Politika Gradyanı (DDPG); deterministik politika gradyanı algoritmasından (DPG) geliştirilmiş, sürekli eylem uzayında, yüksek boyutlu politikaları öğrenmek için derin sinir ağlarından yararlanan, model bağımsız, politika dışı (off-policy), aktör-kritik kullanan bir algoritmadır (Silver ve diğ. 2014; Timothy ve diğ. 2019). Bu algoritma eylem uzayını ayrıklaştırmanın (discretizing) boyutlanma lanetine (curse of dimensionality) yol açmasından dolayı DQN algoritması ile çözülemeyen durumları ele almak için geliştirilmiştir ve deterministik politika gradyanı algoritmasını (DPG) genişletir (Mnih ve diğ. 2015). Bellman denklemi ve Q-öğrenme bu algoritmanın temel bileşenidir. Algoritma eşzamanlı olarak bir Q-değeri fonksiyonunu ve bir politikayı öğrenir. Q-değeri fonksiyonunu öğrenmek için politika dışı verileri ve Bellman denklemini kullanır aynı zamanda politikayı öğrenmek için Q-değeri fonksiyonunu kullanır. Pekiştirmeli öğrenmede eğer optimal eylem-değer fonksiyonu $Q^*(s,a)$ (bkz. (2.5)) biliniyorsa herhangi bir verilen durumda optimal eylem $a^*(s)$ eşitlik (2.7) çözülerek bulunabilir.

$$a^*(s) = arg\max_a Q^*(s,a) \tag{2.7}$$

Eğer ayrık zamanlı eylemler sonlu ise buradaki maksimum değeri hesaplamak sorun teşkil etmemektedir. Çünkü Q-değerleri her bir eylem için ayrık olarak hesaplanabilir. Ancak eylem uzayı sürekli olduğunda (continuous action space) her adımda hesaplaması gerekecektir ve bu işlem kullanılamaz.

Eylem uzayı sürekli olduğunda $Q^*(s,a)$ fonksiyonunun eyleme (a) göre türevi alınabilir olduğu varsayılır. Bu durumda $\max_a Q(s,a) \approx Q(s,\pi(s))$ ile yaklaşık olarak ifade edilen bir politika $\pi(s)$ için etkili bir gradyan tabanlı öğrenme kuralı kullanılabilir. DDPG aynı anda bir Q-değer fonksiyonu $Q(s,a)$ ve bir politika $\pi(s)$ öğrenir. Politika fonksiyonu, bir durum verildiğinde alınacak eylemi belirlerken; Q-fonksiyonu, bu eylemin değerini değerlendirir.

Stokastik politikaların aksine deterministik politika, eylemi doğrudan politika fonksiyonu $\pi(s)$ tarafından belirler. DDPG, aktör politikası $\pi(s|\theta)$ ve kritik Q-değer fonksiyonunu $Q(s,a|\phi)$ güncelleyen aktör-kritik mimarisini kullanır. DDPG dört sinir ağı kullanır; yerel aktör (ϕ), yerel kritik (Q), hedef aktör (ϕ') ve hedef kritik (Q'). Aktör ağları θ parametresini kullanarak politikaya yaklaşmayı hedeflerken eleştirmen ağları (critic networks) ϕ parametrelerini kullanarak Q-değer fonksiyonuna yaklaşır.

DDPG'de kullanılan takviyeli öğrenmedeki iki temel fonksiyon, aksiyon-değer fonksiyonu (bkz. eşitlik (2.2)) ve karşılık gelen Bellman denklemidir (bkz. eşitlik (2.3)). Bellman Denklemi $Q^*(s,a)$'ya yaklaştırıcı öğrenmek için bir başlangıç noktası olarak kullanılır. Eğer politika deterministik ise $\pi : S \leftarrow A$ olarak tanımlanabilir. Farklı bir stokastik davranış politikası β tarafından üretilen geçişi kullanarak Q_ϕ politika dışı (off-policy) olarak öğrenilebilir. Q_ϕ'nin Bellman denklemini sağlamaya ne kadar yaklaştığını yaklaşık olarak bir ortalama kareli Bellman hatası (MSBE) fonksiyonu ile sadece ortama (environment) bağlı olan eşitlik (2.8)'deki gibi oluşturulabilir. Burada d_t, s_{t+1} durumunun terminal olduğunu belirtir. Kayıp (loss) β tarafından üretilen geçişlerden sağlanır. Bu nedenle bu algoritmada tekrar tamponuna ve hedef ağlara büyük önem arz etmektedir (Timothy ve diğ. 2019).

$$L = \frac{1}{M}\sum_{i}\left(y_i - Q(s_i, a_i|\phi)\right)^2$$

$$L(\phi) = \mathbb{E}_{s_t\sim\rho^{\beta}, a_t\sim\beta, r_t\sim E}[(Q(s_t, a_t|\phi) - y_t)^2] \tag{2.8}$$

$$y_t = r(s_t, a_t) + \gamma(1 - d_t)Q'(s_{t+1}, \pi'(s_{t+1}|\theta)|\phi)$$

Başlangıçta aktör ve kritik ağlarının parametreleri rastgele başlatılır. Yerel aktör yani mevcut politika, mevcut duruma dayalı aksiyonlar önererek deneyim tekrarlama tamponunu (replay buffer) doldurmaya başlar. Bu tampon yeterince dolunca, algoritma her zaman adımı t için rastgele bir mini-deneyim (mini-batch) grubu örnekleme işlemine başlar. Bu mini-grup, yerel Q değeri ile hedef Q değeri arasındaki ortalama karesel hatayı (MSE, mean squared error) minimize ederek yerel eleştirmeni güncellemek için kullanılır. Ancak algoritma deneyimlerinin toplu işler halinde politika-dışı bir şekilde politikayı güncellediği için mini-gruptan hesaplanan gradyanların ortalamasını kullanmak mümkündür. Politika perspektifinden bakıldığında amaç denklem (2.9)'u maksimize etmek ve denklem (2.10)'deki politika parametresiyle ilgili hedef fonksiyonunun türevi aracılığıyla politika kaybını hesaplamaktır. Aktör politikası denklem (2.11) ile belirlenen örneklenmiş politika gradyanı kullanılarak güncellenir.

$$\mathcal{J}(\theta) = \mathbb{E}[Q(s, a)|_{s=s_t, a=\pi(s_t)}] \tag{2.9}$$

$$\nabla_{\theta}\mathcal{J}(\theta) \approx \nabla_a Q(s, a)\nabla_{\theta}\pi(s|\theta) \tag{2.10}$$

$$\nabla_{\theta}\mathcal{J}(\theta) \approx \frac{1}{N}\sum_{i}\left[\nabla_{\theta}Q(s, a|\phi)|_{s=s_i, a=\pi(s_i)}\nabla_{\theta}\pi(s|\theta)|_{s=s_i}\right] \tag{2.11}$$

Hedef ağlar, dalgalanmayı önlemek için yumuşak güncelleme mekanizması kullanılarak güncellenir. Bu (2.12) denklemiyle gösterilir ve burada $\varepsilon \ll 1$ küçük bir sabittir. Bu yöntem, yeni bilgilerin yavaş ve kontrollü bir şekilde entegrasyonunu sağlayarak öğrenmenin istikrarını korur.

$$\theta' \leftarrow \varepsilon \cdot \theta + (1 - \varepsilon) \cdot \theta' \tag{2.12}$$

Pekiştirmeli öğrenmede ayrık eylem uzayları için keşif genellikle rastgele bir eylem seçerek yapılır ancak sürekli eylem alanları için ise keşif, eyleme doğrudan gürültü ekleyerek gerçekleştirilir. Timothy ve diğ. (2019), eylem çıktısına gürültü eklemek için Ornstein-Uhlenbeck sürecini kullanmışlardır (Uhlenbeck ve Ornstein 1930). Bu süreç ile $a_t = \pi(s_t|\theta) + N$ şeklinde eylem çıktısına gürültü eklenir. Son olarak eylem istenilen aralıkta kırpılır. DDPG'nin sözde kodu **Şekil 2.3**'de verilmiştir.

Algoritma 2.1: DDPG Algoritması

Girdi: ϕ parametresine sahip ilk eleştirmen ağı Q ve θ parametresiyle aktör ağı π

Q' ve π' hedef ağlarını $\bar{\phi} \leftarrow \phi$, $\bar{\theta} \leftarrow \theta$ ağırlıkları ile başlat

Tekrar tamponu $\mathcal{D}$'yi başlat

for epizod = 1, M **do**

  Eylem keşfi için rastgele bir $\mathcal{N}$ değeri

  $s_t \leftarrow s_1$, İlk gözlem durumunu al. (ortamdan durum bilgileri)

  tekrarla

    $a_t = \pi(s_t|\theta) + \mathcal{N}$, Eylemini seç ve istenen aralıkta kırp

    a_t eylemini sisteme uygula, $s_t, a_t, r_t, s_{t+1}, d_t$ değerlerini $\mathcal{D}$'ye kaydet

    if güncelleme gerekiyorsa

      $\mathcal{D}$'den N geçişinin rastgele minibatchini örnekle

      Hedef fonksiyon y_t'yi hesapla. $r_t + \gamma(1 - d_t)Q'(s_{t+1}, \pi'(s_{t+1}|\bar{\theta})\bar{\phi})$

      Kaybı en aza indirerek kritik ağını güncelle: $L = \frac{1}{M}\sum_i \big(y_i - Q(s_i a_i|\theta)\big)$

      Politikayı örneklenen gradyanı kullanarak güncelle: denklem (2.11)

      Yumuşak güncelleme hedef kritik: $\bar{\phi} \leftarrow \epsilon\phi + (1 - \epsilon)\,\bar{\phi}$

      Yumuşak güncelleme hedef politika: $\bar{\theta} \leftarrow \epsilon\theta + (1 - \epsilon)\,\bar{\theta}$

    sonlandır

    $s_t \leftarrow s_{t+1}$

  s_t terminal **olana kadar**

sonlandır

Şekil 2.3: DDPG algoritmasının sözde kodu.

2.3.2 İkiz Gecikmeli DDPG (Twin Delayed Deep Deterministic Policy Gradient—TD3)

DDPG belirli koşullarda mükemmel performans elde edebilmektedir fakat hiperparametreler ve diğer tuning türlerine göre genellikle zayıf kalmaktadır. DDPG için yaygın bir sorun biçimi, öğrenilen Q-fonksiyonunun Q-değerlerini önemli ölçüde fazla hesaplamaya başlamasıdır; bu da Q-fonksiyonundaki hatalardan yararlandığı için politikanın çökmesine neden olur. İkiz Gecikmeli DDPG (TD3) bu sorunu üç kritik püf noktası sunarak ele alan bir algoritmadır; bunlar kırpılmış ikiz Q-öğrenme, gecikmeli politika güncellemeleri, hedef politikanın yumuşatılmasıdır (Fujimoto ve diğ. 2018).

Kırpılmış ikiz-Q öğrenme; TD3 Q_{ϕ_1} ve Q_{ϕ_2} şeklinde iki farklı Q-fonksiyonunu eş zamanlı öğrenir. Bellman hata kayıp fonksiyonlarındaki hedefleri oluşturmak için iki Q değerinden küçük olanını kullanır. Gecikmeli politika güncellemeleri, TD3 politikayı ve hedef ağları Q-fonksiyonundan daha az sıklıkta günceller. Ayrıca her iki Q-fonksiyonu güncellemesi için bir politika güncellemesi önerilmektedir (Fujimoto ve diğ. 2018). Hedef politikayı yumuşatma için de eylemdeki değişiklikler boyunca Q yumuşatılması ve Q -fonksiyonundan kaynaklı hataların yararlanmasını zorlaştırmak için eyleme gürültü eklenir.

DDPG'de politika $\pi(s|\theta)$ ile belirlenir ancak TD3'te eylem boyutuna göre her bir boyuta kırpılarak gürültü eklenir. Kırpılmış gürültü eklendikten sonra hedef eylem, sisteme uygulanabilir eylem aralığında ($a: a_{alt} \leq a \leq a_{üst}$) kırpılır. Örneğin çalışma kapsamında, direksiyon açısı eylemi -1 ile 1 aralığında olmalıdır. Bu nedenle minimum ve maksimum limitin dışında olduğunda en yakın değere kırpılır (bkz. denklem (2.13)). Bu işlem algoritma için temel olarak bir regülatör işlevi görür. Q-fonksiyonu yaklaştırıcısı bazı eylemlerde ani pikler yaparak kırılgan ve yanlış kararlara sahip olabilir. Bu sayede hedef politika üzerinde yumuşatma yapılarak bunun önüne geçilebilir.

$$a'(s') = clip(\pi(s_{t+1}|\theta) + clip(\epsilon, -c, c), a_{alt}, a_{üst}), \qquad \epsilon \sim \mathcal{N}(0, \sigma) \quad (2.13)$$

TD3'te iki Q-fonksiyonu da tek bir hedef kullanarak 2 farklı hedef değerinden küçük olanı kullanır (bkz. (2.14)). Ardından ikisi de hedef değere eşitlik (2.15)'deki gibi regresyonla öğrenmektedir.

$$y(r, s', d_t) = r + \gamma(1 - d_t) \min_{i=1,2} Q_\phi(s', a'(s')) \tag{2.14}$$

$$\begin{aligned} L(\phi_1) &= \mathbb{E}_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E}[(Q(s_t, a_t|\phi_1) - y_t)^2] \\ L(\phi_2) &= \mathbb{E}_{s_t \sim \rho^\beta, a_t \sim \beta, r_t \sim E}[(Q(s_t, a_t|\phi_2) - y_t)^2] \end{aligned} \tag{2.15}$$

Hedef için küçük Q-değerinin kullanılması ve buna doğru regresyon yapılması, Q-fonksiyonunda aşırı tahmin yapılmasını önlemeye yardımcı olur. Ardından politika Q_ϕ'yi maksimize ederek öğrenilir (bkz. denklem (2.16)).

$$\max_\theta E_{s \sim \mathcal{D}}\big[Q_\phi(s, \pi(s))\big] \tag{2.16}$$

TD3'e ait sözde kod **Şekil 2.4**'te verilmiştir.

Algoritma 2.2: TD3 Algoritması

Girdi: başlangıç politika parametreleri θ, Q-fonksiyon parametreleri ϕ_1, ϕ_2

Q' ve π' hedef ağlarını $\bar{\phi}_1 \leftarrow \phi_1, \bar{\phi}_2 \leftarrow \phi_2, \bar{\theta} \leftarrow \theta$ ağırlıkları ile başlat

Tekrar tamponu $\mathcal{D}$'yi başlat

for epizod = 1, M **do**

  Eylem keşfi için rastgele bir $\mathcal{N}$ değeri

  $s_t \leftarrow s_1$, İlk gözlem durumunu al. (ortamdan durum bilgileri)

  tekrarla

    $a_t = clip(\pi(s_t|\theta) + \epsilon, a_{alt}, a_{\text{üst}})$ $\epsilon \sim \mathcal{N}$, Eylemini seç ve istenen aralıkta kırp

    a_t eylemini sisteme uygula, $s_t, a_t, r_t, s_{t+1}, d_t$ değerlerini $\mathcal{D}$'ye kaydet

    if güncelleme zamanıysa

      $\mathcal{D}$'den N geçişinin rastgele minibatchini örnekle

      Hedef eylemleri hesapla denklem (2.13)

      Hedefi hesapla: denklem (2.14)

      Q-fonksiyonlarını (denklem (2.15)) gradyan iniş kullanarak güncelle:

        Eğer politika geciktirmesi yoksa

          Politikayı örneklenen gradyanı kullanarak güncelle:

$$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi\big(s, \pi(s|\theta)\big)$$

          Yumuşak güncelleme hedef kritik: $\bar{\phi}_{1,2} \leftarrow \epsilon\phi_{1,2} + (1-\epsilon)\bar{\phi}_{1,2}$

          Yumuşak güncelleme hedef politika: $\bar{\theta} \leftarrow \epsilon\theta + (1-\epsilon)\bar{\theta}$

    sonlandır

    $s_t \leftarrow s_{t+1}$

  s_t terminal **olana kadar**

sonlandır

Şekil 2.4: TD3 algoritmasının sözde kodu.

2.3.3 Proksimal Politika Optimizasyonu (Proximal Policy Optimization—PPO)

Proksimal Politika Optimizasyonu (PPO), modern derin pekiştirmeli öğrenme algoritmaları arasında önemli bir yere sahiptir ve özellikle yüksek boyutlu ve sürekli eylem uzaylarında başarılı sonuçlar elde etmektedir (Andrychowicz ve diğ. 2020).

Politika gradyan yöntemlerinde sık karşılaşılan bir sorun, bir parametre güncelleme adımından sonra politika performansının sapması veya çökmesidir. PPO politika optimizasyonunda kararlılığı ve verimliliği artırmak amacıyla geliştirilmiştir. Politikaların güncellenmesi sırasında büyük değişiklikler yapılması, öğrenmenin dengesiz ve verimsiz olmasına neden olabilir. PPO politika güncellemelerini yumuşatarak bu problemi çözmeyi hedefler. Schulman ve diğ. (2017b), PPO algoritmasını, her parametre güncellemesinin bir önceki parametre yinelemesinin güven bölgesi içinde kalmasını sağlamak için ilkeli bir optimizasyon prosedürü olarak önermiştir.

PPO-clip fonksiyonu denklem (2.17)'deki gibi formüle edilmiştir. Burada L'nin açılımı denklem (2.18)'de verilmiştir. Hedefi maksimize etmek için birden fazla stokastik gradyan iniş adımı kullanır.

$$\theta_{k+1} = \arg\max_{\theta} \underset{s,a\sim\pi_{\theta_k}}{E} [L(s,a,\theta_k,\theta)] \tag{2.17}$$

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_k}}(s,a)\right) \tag{2.18}$$

Burada $\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}$, önceki politika yinelemesinin önerilen politika güncellemesine olan olasılık oranını gösterir. ϵ ise en son politika güncellemesinin olasılık uzayında ne kadar değişebileceğine dair bir sınırlama getiren, yanlış bir politika güncellemesi olasılığını azaltan kırpma parametresidir. Bu hedefin basitleştirilmiş hali denklem (2.19)'da verilmiştir. Denklemde g'nin iki durumu vardır (bkz. eşitlik (2.20)).

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a), g\left(\epsilon, A^{\pi_{\theta_k}}(s,a)\right)\right) \tag{2.19}$$

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & A \geq 0 \\ (1-\epsilon)A & A < 0 \end{cases} \tag{2.20}$$

Avantajın negatif ve pozitif olmasına göre bu durumlar şekillenir. Eğer eylem ve durum için avantaj pozitif ise bu durumun objektife katkısı denklem (2.21)'deki gibi olur. Avantaj pozitif olduğunda eylem olasılığı $\pi_\theta(a|s)$ artar bu sayede objektif de artar. Bu denklemdeki min objektifin ne kadar artabileceğine limit koyarak sınırlar. $\pi_\theta(a|s) > (1+\epsilon)\pi_{\theta_k}(a|s)$ durumu sağlandığında $(1+\epsilon)A^{\pi_{\theta_k}}(s,a)$ terimi tavan değerine ulaşır. Sonuç olarak yeni politika eski politikadan uzaklaşamaz. Eğer durum-eylem için avantaj negatif ise objektife katkısı denklem (2.22)'deki gibi olur. Bu nedenle eylem olasılığı daha az olursa objektif artar. Ancak bu terimdeki maksimum hedefin ne kadar artabileceğini sınırlayarak limitler. $\pi_\theta(a|s) < (1-\epsilon)\pi_{\theta_k}(a|s)$ durumu geçerli olduğunda ise maksimum limit sınırlanır ve $(1-\epsilon)A^{\pi_{\theta_k}}(s,a)$ tavan değerine ulaşır. Dolayısıyla tekrar yeni politika bir öncekinden fazla uzaklaşamaz.

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, (1+\epsilon)\right)A^{\pi_{\theta_k}}(s,a) \tag{2.21}$$

$$L(s,a,\theta_k,\theta) = \max\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, (1-\epsilon)\right)A^{\pi_{\theta_k}}(s,a) \tag{2.22}$$

Yukarıda belirtilen saptamalardan kırpma (clipping) metodunun bir düzenleyici olarak hareket ettiğini, politikanın radikal ve keskin bir şekilde değişmesi için teşvikleri ortadan kaldırdığını göstermektedir. Burada ϵ hiperparametresi yeni politikanın eski politika ile arasındaki güncelleme oranını ayarlar. **Şekil 2.5**'te PPO'nun algoritması yer almaktadır.

Algoritma 2.3: PPO Algoritması

Girdi: başlangıç politika parametreleri θ, Q-fonksiyon parametreleri ϕ

1 Tekrar tamponu $\mathcal{D}$'yi başlat

2 **for** epizod = 1, M **do**

3 Ortamda $\pi_k = \pi(\theta_k)$ ile politikayı çalıştır ve bir yörünge seti $\mathcal{D}_k$ topla

4 Kazanılacak ödülleri hesapla: $\hat{R}_t'$

5 Mevcut değer fonksiyonu V_{ϕ_k}'ye dayalı olarak avantaj tahminleri $\hat{A}_t$'yı hesapla

6 PPO-Klip objektifini maksimize ederek politikayı güncelle:

$$\theta_{k+1} \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \epsilon \mathcal{D}_k} \sum_{t=0}^{T} L\left(s_t, a_t, \theta_{k_t}, \theta\right)$$

Genellikle Adam stokastik gradyan artışı metoduyla

7 MSE regresyonuyla değer fonksiyonunu uydur (gradyan iniş yöntemi)

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \epsilon \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\theta(s_t) - \hat{R}_t\right)^2$$

8 **sonlandır for**

Şekil 2.5: PPO algoritmasının sözde kodu.

3. AÇIKLANABİLİR YAPAY ZEKA (EXPLAINABLE ARTIFICIAL INTELLIGENCE—XAI)

Bu bölümde tezin temellerini oluşturan açıklanabilir yapay zeka (Explainable Artificial Intelligence—XAI) hakkında ayrıntılı bilgiler sunulmuştur.

Derin sinir ağları (DNN) bilgisayarla görme, konuşma tanıma, doğal dil işleme ve diğer birçok alanda büyük başarılar elde etmiştir. Bu ağlar birçok önceki makine öğrenmesi tekniğini geride bırakmış ve belirli gerçek dünya görevlerinde en iyi performansı sağlamıştır. DNN'ler tıp, biyoinformatik ve astronomi gibi bilimsel alanlarda da güçlü araçlar haline gelmiştir. Bu alanlar genellikle büyük veri setleri içerir ve derin öğrenme bu verileri işlemek için etkili bir yol sunar. Derin öğrenme modellerinin kara kutu doğası, onların nasıl çalıştığını anlamayı zorlaştırır. Derin öğrenme modellerinin başarılı performanslarına rağmen karmaşık ve anlaşılması zor yapıları; güven ve güvenilirlik, etik ve yasal sorunlar oluşturmaktadır (Zhang ve diğ. 2020). Özellikle hukuk, sağlık ve askeri operasyonlar gibi alanlarda, YZ'nin potansiyel hatalarını anlamak ve bu sistemlere duyulan güveni doğru kalibre etmek kritik öneme sahiptir (Tomsett ve diğ. 2020).

XAI, yapay zeka modellerinin işleyişini açık ve anlaşılır hale getirmek için kullanılan yöntemler ve teknikler kümesidir. Modellerin nasıl çalıştığını, kararlarını nasıl verdiğini ve hangi faktörlerin etkili olduğunu anlaşılmasını sağlar (Barredo Arrieta ve diğ. 2020). XAI yöntemleri; kullanıcıların bu karmaşık ağlarının çıktılarını anlamasını, bunlara güvenmelerine yardımcı olmak için insan tarafından okunabilir açıklamalar sağlamayı hedeflemektedir. Avrupa Birliği düzenlemesi olan Genel Veri Koruma Tüzüğü (GDRP) gibi yönetmelikler; güvenilir XAI çözümlerini genişletmek için önemli etik, meşruyet, güven ve bayes gereksinimi doğuran ileri XAI araştırmalarını yönlendirmek için uygulamaya konulmuştur (Voigt ve von dem Bussche 2017).

Literatürde XAI ile ilgili terimler aşağıda tanımlanmıştır.

- Anlaşılabilirlik (understandability): Bir modelin iç yapısını veya algoritmik işlemlerini açıklamaya gerek kalmadan, işleyişini insanlara anlaşılır kılma özelliğidir.

Kullanıcıların, modelin nasıl çalıştığını anlamasını sağlamayı amaçlar (Montavon ve diğ. 2018).

- Anlaşılırlık (comprehensibility): Bilgiyi insanlar açısından anlaşılır kılarak ifade edebilme.
- Yorumlanabilirlik (interpretability): Bir modelin anlamını insanlara anlaşılır terimlerle açıklama veya sunma yeteneğidir. Genellikle modelin şeffaflığı olarak da ifade edilir ve bir modelin kendi başına anlaşılabilir olma derecesini belirtir (Barredo Arrieta ve diğ. 2020).
- Açıklanabilirlik (explainability): Modelin işleyişini net veya kolay anlaşılır hale getiren detaylar veya nedenler sunmasıdır. Bir yapay zeka modelinin insan kullanıcılar tarafından anlaşılmasını sağlamaya yönelik adımları veya prosedürleri ifade eder (Guidotti ve diğ. 2018).
- Şeffaflık (transparency): Bir model kendi başına anlaşılabilir olduğunda şeffaf olarak kabul edilir. Kara kutu bir modelin tam tersini ifade eder (Kök ve diğ. 2023).

Bu tanımlamalarda anlaşılabilirliğin en temel kavram olarak ortaya çıktığı görülmektedir. Özellikle şeffaflık ve yorumlanabilirlik buna doğrudan bağlıdır.

XAI post-hoc veya ante-hoc, model-agnostik veya model spesifik, global veya lokal açıklamalar olarak kendi içinde sınıflandırılabilir. Post-hoc ve ante-hoc eğitim sırasındaki ve eğitim sonrasındaki açıklama sağlamasıyla ilgilidir. Modele özgü yöntemler genellikle kara kutu modeller olarak da bilinen DNN'ler için kullanılır (Barredo Arrieta ve diğ. 2020). Model agnostik yöntemler ise model sınıfından bağımsız her tür modele uygulanabilir. Model üstünde tekil örnekler ile uygulanan teknikler yerel, tüm modelin anlaşılabilirliğini sağlamak için uygulanan yöntemler ise global olarak tanımlanır. Global açıklamalarda hangi özelliklerin önemli olduğu ve özellikler arasındaki ilişkiler sunulur.

Post-hoc açıklanabilirlik, tasarım gereği şeffaf olmayan modellerin anlaşılabilirliğini artırmak için kullanılan çeşitli tekniklerdir. Bu teknikler modelin çıktılarını ve karar süreçlerini açıklamak için farklı yaklaşımlar kullanır. Metin açıklamaları (text explanations), modelin sonuçlarını ve işleyişini metin tabanlı açıklamalarla ifade eder. Görsel açıklamalar (visual explanations), modelin davranışını ve iç ilişkilerini görselleştirme teknikleri kullanarak açıklamaları içerir.

Örneğin ısı haritaları, özelliklerin önemini ve modelin hangi bölgelerde karar verdiğini görselleştiren tekniklerdir. Yerel açıklamalar (local explanations), modelin sadece belirli alt alanlarını açıklayarak, daha karmaşık çözüm alt alanlarını anlaşılır hale getirir. Örneklerle açıklamalar, modelin çıktısını açıklamak için benzer örnekler kullanarak açıklamayı içerir. Bir modelin belirli bir kararı nasıl verdiğini göstermek için benzer kararların örneklerinden yararlanmak buna bir örnektir. Basitleştirme ile açıklamalar, karmaşık bir modelin işleyişinin daha basitleştirilmiş bir modelle açıklanmasını içerir. Özelliklerin önemi açıklamaları, modelin çıktısını etkileyen özelliklerin önemini hesaplama ve bu önem derecelerini ifade etmektedir. Örneğin SHAP değerleri, modelin hangi özelliklere ne kadar önem verdiğini gösterir (Barredo Arrieta ve diğ. 2020).

XAI'nın kullanıldığı önemli alanlardan bazıları sağlık, hukuk, finans, siber güvenlik, askeri sistemlerdir. Sağlık alanında yapay zeka modelleri; tıbbi görüntülerin analizi, hastalık teşhisi ve tedavi önerileri gibi çeşitli alanlarda yaygın olarak kullanılmaktadır. Bu tür modellerin kararlarını anlamak ve güvenilirliklerini artırmak amacıyla, XAI teknikleri büyük önem taşımaktadır. Örneğin bir YZ modelinin kanser teşhisinde hangi özelliklere dayanarak karar verdiğini açıklamak, doktorların bu kararları değerlendirmesine ve modelin sunduğu teşhis ve tedavi önerilerine güvenmesine yardımcı olur. Böylece sağlık profesyonelleri modellerin karar süreçlerini daha iyi anlayabilir ve gelişmesinde katkı sağlayarak bu teknoloji daha etkin bir şekilde kullanabilir (Albahri ve diğ. 2023; Kurshan ve diğ. 2022). Hukuk alanında yapay zeka modelleri, yasal belgelerin analizi ve hukuki kararların verilmesinde kullanılmaktadır. XAI bu modellerin kararlarını şeffaf hale getirerek, yasal süreçlerde adil ve tarafsız kararlar alınmasını sağlar. Bir YZ modelinin bir davada hangi yasal gerekçelere dayanarak bir öneri sunduğunu açıklamak, avukatların ve hakimlerin bu öneriyi değerlendirmesine yardımcı olur. Bu şekilde hukuki süreçlerde YZ modellerinin daha güvenilir ve şeffaf bir şekilde kullanılabilmesi mümkün olmaktadır (Berk ve Bleich 2013; Górski ve Ramakrishna 2021; Mardaoui ve Garreau 2020). Finans sektöründe yapay zeka modelleri; kredi risk değerlendirmesi, yatırım analizleri ve dolandırıcılık tespiti gibi alanlarda kullanılmaktadır. XAI bu modellerin hangi verileri kullanarak ve hangi analizleri yaparak karar verdiklerini açıklayarak, finansal kararların daha güvenilir ve şeffaf olmasını sağlar. Örneğin bir YZ modelinin, bir müşterinin kredi başvurusunu neden

onayladığını veya reddettiğini açıklaması, bankaların ve müşterilerin bu kararları anlamasına yardımcı olur. Böylece finansal hizmetlerin daha adil ve şeffaf bir şekilde sunulmasını sağlanır (Carta ve diğ. 2021; Chromik ve diğ. 2021; Mandeep ve diğ. 2022). Tüm bu bahsedilen alanlarda modellerin aldığı yanlış kararların açıklanması da modelin eksiklerinin görülmesine fayda sağlayacaktır.

3.1 DRL ve Otonom Sistemler için XAI

DRL modellerinin karmaşık karar alma problemlerinde, büyük başarılar elde ettiğine bir önceki bölümde değinilmiştir. Ancak bu modellerin kara kutu doğası, anlaşılmalarını zorlaştırır. Bu nedenle XAI teknikleri, modellerin karar süreçlerini şeffaf hale getirmek için kullanılır.

Iyer ve diğ. (2018), nesne tanıma ve sınıflandırma problemi için açıklanabilir ve nesne duyarlı bir DRL modeli önermiştir. Modelde insan tarafından anlaşılabilir görselleştirmeler sağlamak için "nesne belirginlik haritaları" kullanılmıştır. DRL modellerinin durumlarını ve eylemlerini görselleştirerek, kullanıcıların modellerin nasıl çalıştığını anlamalarına yardımcı olur.

Nahata ve diğ. (2021) ve arkadaşları, dinamik ortamlarda çarpışma riskini değerlendirip açıklamak için XAI tekniklerini kullanmıştır. Bu çalışmada, gerçek dünya sürüş verileri kullanılarak çarpışma risklerinin değerlendirilmesi ve açıklanması için SHAP yöntemleri kullanılmıştır.

Otonom araçlar çevresel verileri sürekli olarak değerlendirir ve karmaşık kararlar alır. Verilen kararlar yoldaki diğer araçlarla, yayalarla etkileşimleri ve trafikteki ani değişikliklere verilen tepkileri içerebilir. Otonom sistemler, karmaşık çevrelerde bağımsız olarak hareket edebilir ve güvenilirlik ile güvenlik açısından yüksek bir standart gerektirir. XAI uygulaması, alınan kararların altında yatan nedenleri ve seçilen hareket tarzlarını aydınlatarak hem sektör profesyonellerine hem de son kullanıcılara araç davranışları üzerinde daha fazla bilgi ve denetim sağlar. Ayrıca hukuki ve mevzuata uygunluk açısından da hesap verilebilirlik sağlar. Otonom araçlar ve XAI hakkında daha ayrıntılı bilgi Bölüm 4'te İlgili Çalışmalar alt başlığında işlenmiştir.

3.2 Tezde Kullanılan Açıklanabilirlik Yöntemleri

Bu alt başlıkta tez kapsamında kullanılan XAI yöntemleri detaylandırılmıştır. Kullanılan başlıca yöntemler arasında SHAP (SHapley Additive exPlanations) içerisinde bulunan özet grafikler, özelliklerin ısı haritası grafiği ve kuvvet grafiği (force plot) gibi grafikler yer almaktadır. Ayrıca eğitim esnasındaki kaza yapılan bölgelerin ısı haritası ve kaza bilgilerinin istatistikleri incelenmiştir. Eğitim bölümünde ayrı olarak değerlendirilen bu veriler, daha sonra eğitilmiş modeller ile test yarışları yaptırılarak aksiyon-zaman grafiklerine dönüştürülmüştür.

Greydanus ve diğ. (2017), DRL ajanlarının davranışlarını anlamak amacıyla belirginlik haritalarını (saliency maps) kullanmıştır. Atari 2600 ortamlarını kullanarak yapılan bu çalışmada ajanların neye dikkat ettikleri, doğru veya yanlış nedenlerle karar verip vermedikleri ve öğrenme sürecinde nasıl evrildikleri incelenmiştir. Araştırma pertürbasyon tabanlı bir teknik ile yararlı belirginlik haritaları oluşturmuş ve bu haritaların, ajanların karar alma süreçleri hakkında önemli içgörüler sağladığını göstermektedir. Ayrıca belirginlik haritalarının, insanlar tarafından anlaşılabilir açıklamalar sunarak, DRL ajanlarının anlaşılmasını kolaylaştırdığı bulunmuştur.

Heuillet ve diğ. (2020), DRL algoritmalarının açıklanabilirliğini artırmak amacıyla SHAP ve belirginlik haritaları gibi post-hoc açıklanabilirlik tekniklerini incelemiştir. SHAP modelin her bir özelliğinin karar üzerindeki katkısını açıklarken, belirginlik haritaları bir görüntünün hangi bölümlerinin model için önemli olduğunu vurgulamaktadır. SHAP ve belirginlik haritaları kullanılarak DRL modellerinin karar alma süreçlerinin nasıl daha şeffaf hale getirilebileceği incelenmiştir. Özellikle SHAP yönteminin, DRL modellerinin kararlarını anlamak için genel bir çerçeve sunduğu ve farklı ortamlarda ve modellerde uygulanabilir olduğu belirtilmektedir. Belirginlik haritaları ise DRL ajanlarının belirli görevler sırasında hangi bilgiye odaklandığını görselleştirmede kullanılmıştır.

SHAP, model tahminlerini açıklamak için oyun teorisinden yararlanan güçlü bir tekniktir. SHAP her bir özelliğin belirli bir tahmine ne kadar katkıda bulunduğunu hesaplayarak, modelin karar verme sürecini anlaşılır hale getirir. SHAP değerleri (SHAP values) her bir özelliğin önem derecesini ve bu özelliklerin model çıktısına olan pozitif veya negatif etkilerini/katkılarını gösterir. Bu yöntem derin öğrenme

modelleri gibi karmaşık kara kutu modellerin iç işleyişini açığa çıkararak, kullanıcıların modelin nasıl çalıştığını ve neden belirli kararlar aldığını anlamalarını sağlar. Lundberg ve Lee (2017) tarafından sunulan bu yaklaşım, çeşitli model-agnostik ve modele özgü yöntemlerle birleştirilerek geniş bir uygulama yelpazesi için kullanılabilir hale getirilmiştir.

SHAP yönteminin temelinde bir özelliğin model çıktısı üzerindeki etkisini ölçmek için beklenen değerler arasındaki farklar kullanılır. Özellikle SHAP değerleri, belirli bir özellik bilindiğinde model çıktısının beklenen değeri ile bu özellik bilinmediğinde model çıktısının beklenen değeri arasındaki fark olarak tanımlanır. Bu her özelliğin model tahminine ne kadar katkıda bulunduğunu gösterir. SHAP yöntemi hem model agnostik hem de modele özgü yöntemler sunar; yani herhangi bir model türüne uygulanabilir ve belirli model türleri için optimize edilebilir. Örneğin derin öğrenme modelleri için Deep SHAP ve Kernel SHAP gibi varyantlar kullanılır.

Kernel SHAP (Linear LIME + Shapley değerleri): SHAP değerlerini hesaplamak için model-agnostik bir yaklaşım kullanır ve SHAP değerlerinin hesaplanmasını optimize eder. Bu yöntem klasik Shapley değeri denklemlerinin bir permütasyon versiyonunun örnekleme ile yaklaştırılmasını sağlar.

Deep SHAP (DeepLIFT + Shapley değerleri): SHAP değerlerini derin öğrenme modelleri için hesaplamak üzere tasarlanmış bir yöntemdir. Bu yöntem DeepLIFT'in, SHAP değerlerine dayalı olarak derin ağların bileşenlerini çözümlemesi ve birleştirmesi prensibi ile çalışır. Deep SHAP, derin ağların her bir bileşeni için SHAP değerlerini hesaplayarak bu değerleri bütün model için SHAP değerlerine dönüştürür.

SHAP kuvvet grafikleri (force plots), belirli bir tahminin arkasındaki karar süreçlerini görselleştirmek için kullanılır. Bu grafikler modelin belirli bir tahmine nasıl ulaştığını ve her bir özelliğin bu tahmin üzerindeki etkisini gösterir. Özellikler modelin tahminini artıran ve azaltan faktörler olarak ikiye ayrılır ve genellikle renklerle kodlanır (örneğin tahmini artıran özellikler kırmızı, azaltanlar ise mavi ile gösterilir). Böylece kullanıcının modelin hangi özelliklerden dolayı belirli bir sonuca ulaştığını daha kolay anlamasını sağlar. SHAP kuvvet grafikleri özellikle sağlık alanında model tahminlerini açıklamak için kullanılır. Örneğin bir hastanın hastalık riski tahmin

edilirken, hangi klinik özelliklerin bu riski arttırdığını veya azalttığını gösterebilir (Lundberg ve diğ. 2018).

SHAP yöntemi çeşitli karmaşık sistemlerde ve DRL modellerinde, karar süreçlerini açıklamak için farklı çalışmalarda kullanılmıştır. He ve diğ. (2021), otonom bir UAV'nin simüle edilmiş bir ortamda gezinmesini sağlamak için SHAP'ı kullanmışlardır. SHAP değerlerini kullanarak UAV'nin kinematik durumlarının veya CNN katman aktivasyon haritalarının katkı değerlerini hesaplamışlardır. Bu sayede UAV'nin kararlarını etkileyen en önemli ağ bölümlerini belirleyebilmişlerdir. Remman ve diğ. (2021), bir robotik kaldıraç manipülatör sisteminin eylemlerini açıklamak için SHAP'ı kullanmışlardır. Kernel SHAP yöntemini kullanarak, bir özelliğin çıktı üzerindeki doğrudan ve dolaylı etkilerini tanımlayan Causal SHAP'ı geliştirmişlerdir. Bu yöntem, robotik manipülatörlerin kararlarını daha iyi açıklamıştır. Chang Wang ve diğ. (2020) ise SHAP yöntemini kullanarak bir otomatik vinç sistemini açıklamışlardır. Bu çalışma vinç yükünün hareketini kontrol ederek, yük salınımını minimumda tutmak için DRL ajanının nasıl çalıştığını görselleştirmiştir.

Bu örnekler SHAP'ın karmaşık sistemlerde nasıl kullanıldığını ve bu sistemlerin karar süreçlerini nasıl daha anlaşılır hale getirdiğini göstermektedir. Bu çalışma kapsamında da DRL ile otonom sürüşün açıklanabilirliğinde bu yöntemlerden yararlanılmıştır.

4. DRL ALGORİTMALARINA DAYALI OTONOM SÜRÜŞ İÇİN XAI UYGULAMALARININ GELİŞTİRİLMESİ

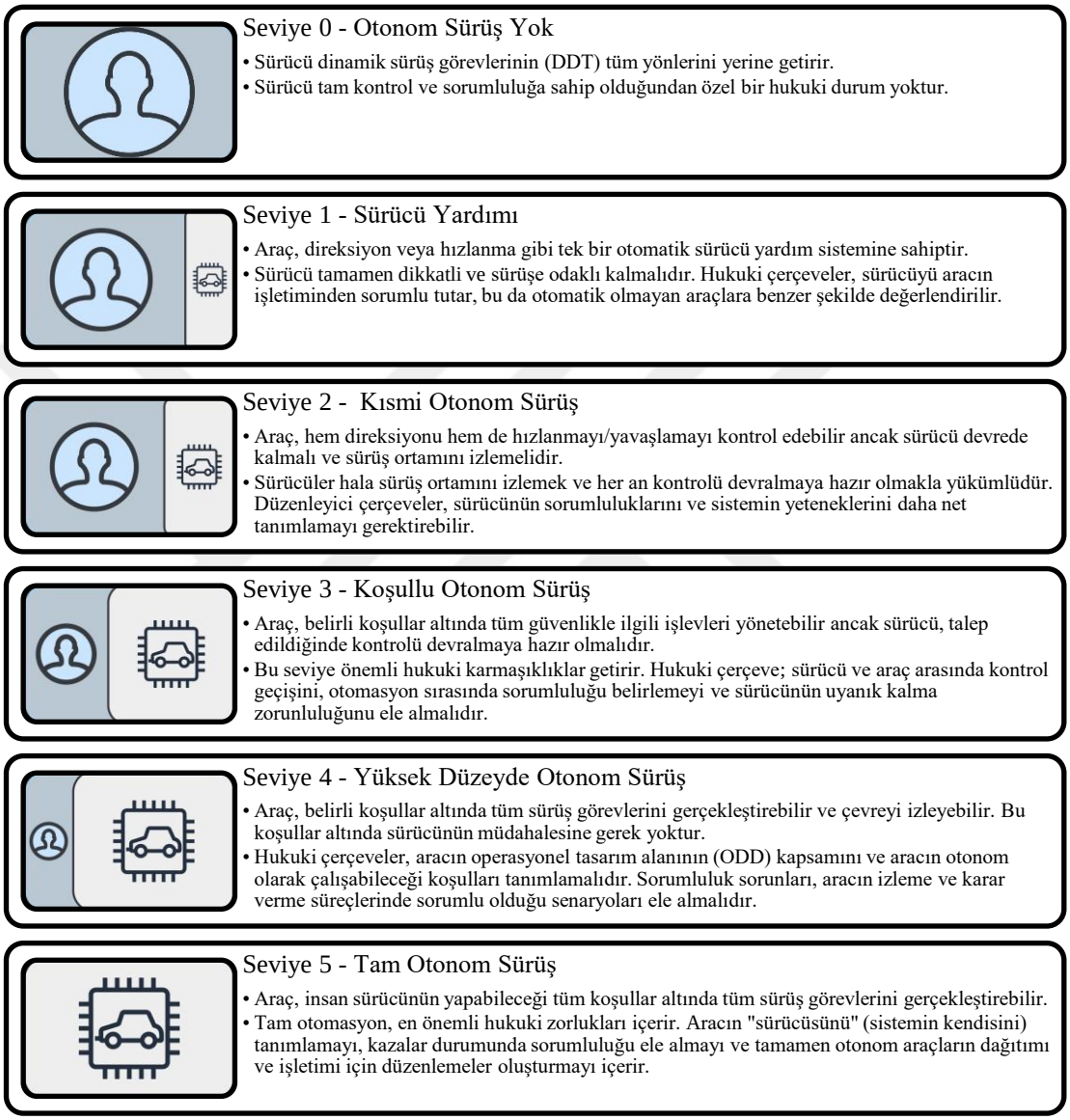
Bu bölüm DRL algoritmalarına dayalı otonom sürüş sistemleri için XAI uygulamalarının geliştirilmesi kapsamında; simülasyon ortamı, model eğitimi, modellerin test senaryoları ve kullanılan XAI yöntemleri hakkında bilgi sunmaktadır. Ayrıca problem tanımı yapılmış, uygulamanın önemi vurgulanmış ve ilgili çalışmalardan bahsedilmiştir. Öncelikle otonom araçlarda karşılaşılan problemin tanımı, bu problemlerin çözümünde XAI'nın önemi ve ihtiyaçları tartışılmıştır. Ardından literatürdeki ilgili çalışmalar ve bu çalışmaların eksiklikleri ele alınarak; TORCS simülasyon ortamının özellikleri, kurulumu ve yapılandırılması detaylı bir şekilde açıklanmıştır. Bu bölüm DRL algoritmalarının eğitimi ve stratejilerinin yanı sıra, otonom araçlarda XAI uygulamalarının nasıl geliştirilebileceğine odaklanmaktadır.

4.1 Problem Tanımı

Bu alt başlıkta otonom sürüş ve XAI'daki açıklanabilirlik, hesap verebilirlik, yasal yükümlülük problemleri açıklanmış ve tez kapsamında probleme yaklaşılan çözüme odaklanılmıştır.

Otonom sürüş, araçların insan müdahalesi olmadan hareket etmesini sağlayan gelişmiş teknolojilerle donatılmış bir ulaşım otomasyonu biçimidir. Çevreyi algılamak, karar almak ve sürüş işlevlerini yerine getirmek için derin öğrenme yöntemleri, kontrol algoritmaları, sensörler ve kameralar kullanılır. Otonom sürüş, Otomotiv Mühendisleri Topluluğu (SAE) (2021) tarafından Seviye 0 (otomasyon yok) ile Seviye 5 (tam otomasyon) arasında değişen ve bir aracın geleneksel olarak bir insan sürücü tarafından gerçekleştirilen görevleri devralma derecesini gösteren farklı seviyelerde sınıflandırılmaktadır. Bu seviyeler ile ilgili bilgi Şekil 4.1'de verilmiştir. Şekilde yer alan seviyelere göre seviye 2'ye kadar sürüş görevinden insan sürücü sorumludur. Seviye 3'te sürücü, belirli koşullar altında sürüş görevinden uzaklaşabilir ancak sistemi izlemek zorundadır. Kaza durumunda sürücü, araç sahibi ve üretici arasında sorumluluk paylaşılır. Seviye 0'da sürücü hem boylamasına hem de yanal

yönlendirmeden sorumludur. Aktif bir müdahaleci araç sistemi yoktur. Seviye 1'de sürücü her zaman boylamasına veya yanal yönlendirmeden sorumludur. Sistem, bazı sürüş yardımlarını üstlenir. Sürücü sistemi her zaman izlemek zorundadır. Sistem, belirli bir kullanım durumu içinde boylamasına ve yanal yönlendirmeyi üstlenir.



Şekil 4.1: Otonom sürüş için seviyeler.

Bir sonraki seviyede sürücü sistemi her zaman izlemek zorunda değildir. Sistem, belirli bir kullanım durumunda boylamasına ve yanal yönlendirmeyi üstlenir. Ancak sistem sınırlarını tanır ve sürücüden aracı kontrol etmesini isteyebilir. Seviye 4'te belirli bir kullanım durumu için sürücüye ihtiyaç yoktur. Sistem bu durumlarda tüm durumları otomatik olarak yönetebilir. Son seviyede ise başlangıçtan varış

noktasına kadar hiçbir sürücüye ihtiyaç yoktur. Sistem, herhangi bir hızda ve herhangi bir yol türünde tüm sürüş görevlerini yerine getirebilir.

Zhao ve diğ. (2024), otonom sürüş sistemlerinin mimarisini iki ana başlıkta tanımlamıştır: katmanlı (layered) mimari ve uçtan uca (end-to-end, E2E) mimari. Katmanlı mimari otonom sürüş sistemini üç katmana ayırır. Bunlar algı ve lokalizasyon, planlama ve kontrol katmanıdır. Uçtan uca mimari ise doğrudan ham sensör verileri ile sürüş politikalarını öğrenmesini ifade eder. Katmanlı mimarilerde algı ve lokalizasyon katmanında çevrenin dinamik bir temsili oluşturulur. GPS, IMU, kamera ve lidar gibi sensörlerden alınan bilgiler işlenir. Lokalizasyon, haritalama ve sensör birleştirme işlemleri gerçekleştirilir. Planlama katmanı, yol planlaması, güvenlik ve trafik kurallarına uygun davranışları belirlemeyi içerir. Kontrol katmanı ise hareket planlama sisteminden gelen veriler ile aracın planlanan hareketi gerçekleştirmesini sağlar. Uygun direksiyon açısı, fren, gaz gibi gerekli eylemleri hesaplar. Katmanlı mimari modüler yapısı ve güvenilirlikle ön plana çıkmaktadır. Ancak daha fazla geliştirme maliyetine sahiptir ve karmaşıklık içermektedir. Hesaplama maliyeti de yüksektir ve buna bağlı gecikme söz konusu olabilir. Uçtan uca mimaride ise özellik çıkarımı yapmadan, sistemi modüllere ayırmadan bir bütün olarak ele alır. Sistem sürüş politikasını uygularken doğrudan sensör verilerini alır ve buna göre eylem üretir. İşlem zinciri daha basittir ve daha düşük yanıt gecikmesi yaşanır. Bu nedenle verimlidir. Ancak büyük ölçüde eğitim verisi gerektirir ve karar verme süreçleri şeffaf değildir. Bu nedenle kara kutu bir model olarak görülür.

Avrupa Komisyonu'nun (2020) yayımladığı bir rapor, bağlantılı ve otonom araçların geliştirilmesi ve kullanımıyla ilgili 20 öneri sunmuştur. Bağlantılı ve otonom araçların yol güvenliği, mahremiyet, adil kullanım, açıklanabilirlik, sorumluluk gibi konularda öneriler içermektedir. Açıklanabilirlik konusunda kullanıcı merkezli yöntemler ve arayüzler geliştirerek; algoritmaların teknolojisinin işleyişinin şeffaf ve anlaşılır olması gerektiği ifade edilmektedir. Ayrıca sistemlerin yeteneklerinin ve amaçlarının açıkça iletilmesi ve sonuçlarının izlenebilir olması gerektiği üzerinde durulmuştur.

Verilen tavsiyelerde algoritmik kararların şeffaflığının arttırılması önerilmiş, algoritmanın ve YZ'ye dayalı işlemlerin kullanıcı tarafından anlaşılabilir ve denetlenebilir olması gerektiği, açıklamaların bireylerin ve grupların karşılaştığı

olumsuz sonuçları anlamalarını sağlayacak şekilde net ve anlaşılır olması gerektiği belirtilmiştir. Otonom araçların davranışları için hesap verebilirlik (açıklama yükümlülüğü) kavramı da tartışılmıştır. Sistemlerin ve karar mekanizmalarının karmaşıklığı, aktörlerin rolünün ve kararlarının açıklanabilir olması gerektiğini vurgulamıştır. Ayrıca sistem hatalarının sorumluluğunun doğru bir şekilde atanabilmesi ve hatalı kararların düzeltilmesi için açıklanabilirliğin kritik bir öneme sahip olduğunu belirtmiştir (European ve diğ. 2020).

Otonom sürüşte hesap verebilirlik, birden fazla karmaşık algoritma ve çeşitli işlemler nedeniyle zorlu bir konu haline gelir; bu da sorumluluk boşluklarına neden olabilir. Honda, tam otomatik sürüşün gerçekleştirilmesinden sonra bir kazanın meydana gelmesi durumunda sorumluluğun nerede olduğunu açıklığa kavuşturmak için yasal çerçevelerin yürürlüğe konması gerektiğini belirtmiştir (Honda 2021). Mulgan (2000) tarafından tanımlanan hesap verebilirliğin sosyal yönü; önerilen yaklaşımların, eylemlerin neden ve sonuçlarının ilgili paydaşlara anlaşılır bir şekilde iletilebileceği açıklama mekanizmalarına bağlanabilmesini gerektirmektedir. Bu nedenlerden dolayı geliştirilen algoritmaların şeffaflığı, hesap verilebilirliği ile doğrudan ilişkilidir.

Bir diğer önemli konu, otonom araçlara olan güvenin sağlanmasıdır. Bu güvenin sağlanabilmesi için konunun sosyal ve psikolojik boyutlarıyla ele alınması gerekmektedir. Otomasyona güven John D. Lee ve See (2004) tarafından belirsizlik ve kırılganlık içeren durumlarda bir ajanın veya otomasyonun, bireyin hedeflerine ulaşmasına yardımcı olacağına dair sosyal-psikolojik bir kavram olarak tanımlanmaktadır. Bu tanım güvenin sadece teknik bir kavram olmadığını, aynı zamanda sosyal ve psikolojik boyutları da içerdiğini vurgulamaktadır. Güven; otomasyonun kabulünü, kullanımını ve insanlarla olan etkileşimini etkileyen önemli bir faktör olduğu belirtilmiştir. Otomasyon daha karmaşık hale geldikçe ve anlaşılması zorlaştıkça, güvenin rolü daha da önemli hale gelecektir. Bu nedenle, açıklanabilirlik güven sağlamada önemli bir araçtır. Abraham ve diğ. (2017) otonom araçların vaat ettiği büyük potansiyele rağmen tüketicilerin güven algısının hala beklenenden düşük olduğunu ve insanların bu teknolojiye karşı hala tereddütlü olduklarını belirtmişlerdir. Hoffman ve Klein (2017) otonom sürüş için kullanıcılara anlamlı açıklamalar

sağlamanın, bu alanda güven oluşturmak için gerekli olduğunu savunmuştur. Bu nedenlerden ötürü otonom sürüşte açıklamaların sağlanması büyük önem taşımaktadır.

Omeiza ve diğ. (2022) çalışmalarında, otonom sürüş sistemlerinin karar verme süreçlerinin GDPR benzeri düzenlemelere uygun olarak, bir olay öncesi, sırası ve sonrasında açıklanabilir ve sorgulanabilir olması gerektiği vurgulanmıştır. Bu sistemlerin her bir modülü (algı, karar verme, eylem) için ayrı ayrı açıklanabilirlik tanımlanmış ve bu eylemlerin sürekli izlenmesi, araçların farklı koşullardaki davranışlarının değerlendirilmesine olanak tanıyacağı belirtilmiştir. Ayrıca otonom sürüşün karmaşıklığı ve kara kutu niteliğinin, denetimi zorlaştırdığına dolayısıyla güvenilirliğin değerlendirilmesi için çeşitli ara işlemlerden elde edilen verilerin dikkate alınması gerektiğine işaret etmiştir. Bu yaklaşımın güvenilirlik değerlendirmesini daha kapsamlı ve etkili hale getirebileceğine değinilmiştir (Jacovi ve Goldberg 2020).

Avrupa Birliği'nin Genel Veri Koruma Tüzüğü (GDPR) (2017), kişisel veriler üzerinde daha fazla kontrol sağlama ve bu verileri kullanarak alınan kararların mantığını ve muhtemel sonuçlarını açıklama zorunluluğu getirmiştir. Ayrıca otonom araçların ve ilgili teknolojilerin toplanan veriler üzerinden karar verme süreçlerinin şeffaflığı ve anlaşılabilirliği ile ilgili düzenlemeler üzerinde durulmuştur.

Otonom sürüş ile ilgili olarak, Ulusal Ulaşım Güvenlik Kurulu (NTSB) (2018), kaza soruşturmalarının doğru bir şekilde sağlanabilmesi için araçlarda makul ve güvenilir açıklamaların sağlanmasını kolaylaştırabilecek veri kaydı için çağrıda bulunmuştur. Bu düzenlemeler otonom sürüş güvenliğini arttırabilir ve kaza sonrası hukuki süreçleri kolaylaştırabilir.

Özellikle otonom araçların karmaşık karar verme süreçlerini açıklama yeteneğinin artırılması gerektiği anlaşılmaktadır. Bu hem teknik açıdan hem de kullanıcı deneyimi açısından otonom araçların daha iyi anlaşılmasını ve toplum tarafından kabul edilmesini sağlayabilir.

Bu çerçevede otonom araçlarda XAI uygulamaları ile aşağıdaki sorunların giderilmesi hedeflenmektedir.

- Güvenin artırılması: Algoritmaların nasıl ve neden belirli kararlar verdiğini açıklamak, kullanıcıların sistemlere olan güvenini artırır. Güven özellikle kullanıcıların hayatını doğrudan etkileyen teknolojilerde, adaptasyon ve etkileşim düzeyini belirleyen temel bir faktördür.
- Hata analizi: XAI, algoritmaların performansını değerlendirirken hataların veya istenmeyen sonuçların kökenlerini anlamayı kolaylaştırır. Böylece sistem hatalarını düzeltme ve daha güvenli teknolojiler geliştirme kapasitesi sağlar.
- Mevzuata uygunluk: Birçok yargı makamı otonom sistemlerin konuşlandırılması için şeffaflık ve hesap verebilirlik gereklilikleri getirmiştir. XAI, otonom sistemlerin karar verme süreçlerinin belgelenebileceği ve gözden geçirilebileceği bir mekanizma sağlayarak bu düzenlemelere uyulmasında kritik bir rol oynamaktadır. Bu sayede sistemlerin yasal standartlarla uyumlu olmasını ve gerektiğinde denetlenebilmesini sağlar.
- Eğitim ve geliştirme: Sistem geliştiricileri için XAI, algoritmaların zayıf yönlerini ve geliştirilmesi gereken alanları belirlemede önemli bir araçtır. Böylece sürekli iyileştirme ve optimizasyon süreçlerini destekler.

Bu çalışma, DRL algoritmalarına dayalı otonom sürüş için açıklanabilir yapay zeka yöntemleriyle belirtilen problemlere çözüm üreterek yeni bir yaklaşım sunmayı hedeflemektedir. Simülasyon verileri kullanılarak sistemin çeşitli durumlardaki davranışları analiz edilmiş ve değerlendirilmiştir. Eğitim sürecinde kaza yoğunlukları zamana göre değerlendirilmiş ve bu süreçteki davranışlar izlenmiştir. Bu kapsamda, otonom araçlar ve açıklanabilir yapay zeka alanında önemli katkılar sağlanması hedeflenmektedir.

4.2 İlgili Çalışmalar

Bu alt bölümde, DRL'nin otonom sürüşe uygulanmasında son zamanlarda yapılan araştırmaların sağladığı önemli katkılar ele alınmaktadır. Farklı DRL tekniklerini kullanan temel çalışmaları gözden geçirerek metodolojilerini, ortamlarını ve sonuçlarını karşılaştırılmıştır. Bu genel bakış teknolojinin mevcut durumu hakkında kapsamlı bir anlayış sağlamayı ve DRL ile otonom sürüş alanındaki yenilikleri ve zorlukları vurgulamayı amaçlamaktadır.

Otonom sürüş sistemlerinin geliştirilmesi, yapay zeka, özellikle de DRL alanındaki ilerlemelerin etkisiyle son yıllarda önemli ilerlemeler kaydetmiştir. Otonom sürüş sistemlerinin bileşenleri, bir aracın çevresiyle etkileşimde bulunarak güvenli ve etkili bir şekilde yol almasını sağlamak için gerekli olan çeşitli modüllerden oluşur. Bu bileşenler, çevresel algılama, haritalama, yerelleştirme, karar verme ve kontrol gibi temel işlevleri yerine getirir. Ayrıca otonom sürüş görevleri arasında şerit takibi, şerit değiştirme ve sollama, rampa ile yola geçiş, kavşaklardaki davranışlar, hareket yollarının planlanması gibi görevleri de sürüş esnasında gerçekleştirir. Sürüş ortamlarının ve görevlerinin karmaşıklığı arttıkça, dinamik senaryolarda gerçek zamanlı kararlar verebilen gelişmiş kontrol sistemlerine duyulan ihtiyaç da artmaktadır.

DRL'de yapılan son araştırmalar, otonom sürüşün çok yönlü zorluklarını ele almak için olanaklarından yararlanmaya odaklanmıştır. DRL algoritmaları şeritte kalma ve çarpışmadan kaçınmadan sollama ve kavşaklarda gezinme gibi karmaşık davranışlara kadar çok çeşitli görevlerin üstesinden gelmek için kullanılmıştır. DRL'nin temel avantajı, çevreyle etkileşim yoluyla optimum sürüş politikalarını öğrenme ve geri bildirime dayalı olarak performansı sürekli iyileştirme özelliğidir.

Çok sayıda çalışma çeşitli DRL algoritmalarını ve bu algoritmaların otonom sürüşteki uygulamalarını araştırmıştır. Bu çalışmalardan uçtan uca otonom sürüş için sınıflandırma Tablo 4.1'de yapılmıştır. Bu çalışmalar genellikle öğrenme verimliliğini ve sürüş performansını artırmak için uzman gösterimleri, çoklu sensör girdileri ve gelişmiş ödül fonksiyonları gibi unsurları içeren yeni çerçevelerin geliştirilmesini ve değerlendirilmesini içerir. TORCS ve CARLA gibi simülasyon ortamları, otonom sürüş modellerinin eğitimi ve test edilmesi için gerçekçi platformlar sağlayarak bu çalışmalarda önemli bir rol oynamaktadır.

Xia ve diğ. (2016), Filtrelenmiş Deneyimlerle Derin Q-öğrenme (DQFE) adlı iyileştirilmiş bir derin takviye öğrenme çerçevesi geliştirmiştir. TORCS simülatöründe test edilen bu strateji, modeli eğitmek için geçmiş profesyonel sürüş verilerinden yararlanarak öğrenme hızını %71,2 oranında önemli ölçüde artırırken sürüş dengesini de geleneksel yöntemlere kıyasla %32 oranında geliştirmiştir. DQFE modelinin karmaşık sürüş senaryolarını yönetmedeki etkinliği, otonom araç teknolojilerini ilerletme potansiyelini ortaya koymuştur.

Gu ve diğ. (2017), otonom araçlarda yol takibini iyileştirmek için denetimli öğrenmeyi derin takviyeli öğrenme ile entegre eden hibrit bir öğrenme çerçevesi sunmuştur. TORCS ortamında Denetimli DDPG algoritmasını kullanan bu yaklaşım, geleneksel yöntemlere kıyasla gelişmiş öğrenme hızı ve yol tutuşu göstermiştir.

Ye ve diğ. (2017), otonom sürüş algoritmalarının sağlamlığını ve verimliliğini artırmak için tasarlanmış yenilikçi bir DRL çerçevesi önermiştir. Kısıtlı bir Markov Karar Sürecine bir Negatif-Kaçınma Fonksiyonu entegre ederek ve DDPG algoritmasını kullanarak, model geleneksel DRL yaklaşımlarını önemli ölçüde geliştirmiştir. TORCS simülatöründe test edilen hayatta kalma odaklı DDPG, daha hızlı yakınsama ve ödül tasarımına karşı daha fazla duyarsızlık göstererek otonom araçların güvenliği ve güvenilirliğinde ilerleme vaat etmiştir.

Zuo ve diğ. (2017), insan deneyimlerini DDPG ile bütünleştiren yenilikçi bir pekiştirmeli öğrenme yaklaşımı önermiştir. Bu yöntem otonom sürüş eylemlerini insan benzeri tercihlerle uyumlu hale getirmekle kalmamış, aynı zamanda öğrenme sürecinin kararlılığını ve verimliliğini de artırmıştır.

Chishti ve diğ. (2018), görsel veri işleme için CNN ve karar verme için DQN kombinasyonunu kullanarak otonom araçları eğitmek için yeni bir yaklaşım sunmuştur. Araştırma özel bir simülasyon ortamında eğitilen Raspberry Pi tabanlı bir aracın otonom olarak trafik koşullarında gezinmesine ve yanıt vermesine odaklanmıştır. CNN'i Q-öğrenme ile entegre ederek, araç farklı yol işaretlerini ve sürüş senaryolarını tanımayı ve bunlara uygun şekilde tepki vermeyi öğrenerek, insan müdahalesi olmadan güvenli ve etkili bir şekilde çalışma yeteneğini geliştirmiştir.

Fayjie ve diğ. (2018), karmaşık kentsel senaryolarda otonom araç navigasyonunu geliştirmek için DQN uygulamasını araştırmıştır. DRL'nin gelişmiş duyusal girdileri ve öğrenme algoritmalarını entegre ederek insan seviyesinde sürüş performansı elde etme potansiyelini göstermiştir.

Hilleli ve El-Yaniv (2018), simüle edilmiş ortamlar yerine ham görsel girdiler kullanarak otonom sürüş sistemlerini eğitmek için yeni bir yaklaşım sunmuştur. Sistem, taklit öğrenmeyi bir Çift Derin Q-Ağı aracılığıyla derin takviyeli öğrenme ile birleştirerek, bir yarış oyununda yalnızca görsel ipuçlarına dayalı olarak bir aracı

yönlendirmeyi öğrenmiştir. Bu yöntem, geleneksel simülasyon tabanlı eğitimin mümkün olmadığı gerçek dünya senaryolarında derin öğrenme tekniklerini doğrudan uygulama potansiyelini göstermiştir.

Liang ve diğ. (2018), karmaşık kentsel ortamlarda sürüş politikalarını optimize etmek için taklitçi öğrenmeyi DDPG ile entegre eden yeni bir derin takviye öğrenme çerçevesi önermiştir. Bu yaklaşım, insan benzeri sürüş gösterileriyle keşfe rehberlik ederek örnek verimliliğini artırmakla kalmamış, aynı zamanda farklı kentsel senaryolar arasında politika genellemesini de geliştirmiştir. CARLA simülatöründe test edilen CIRL, geleneksel yöntemlere kıyasla üstün performans ve uyarlanabilirlik göstererek otonom sürüş araştırmaları için yeni bir ölçüt oluşturmuştur.

K. Liu ve diğ. (2018), TORCS simülatöründe otonom sürüş görevlerini öğrenmenin hızını ve istikrarını toplu olarak iyileştiren uzman gösterileri ve yeni bir denetimli kayıp işlevi ekleyerek DDPG çerçevesini geliştirmiştir. Bu yaklaşım, eğitim verimliliği ve politika kararlılığındaki önemli zorlukları ele almıştır.

Luo ve diğ. (2018), otonom sürüş uygulamalarındaki genelleme sorunlarını ele almak için yeni bir derin takviye öğrenme çerçevesi olan Orthogonal Policy Gradient Descent'i (OPGD) tanıtmışlardır. TORCS ortamında test edilen OPGD, politika gradyanlarını Q değerlerine karşı ortogonal olarak optimize ederek üstün performans göstermiş ve geleneksel yöntemlere kıyasla daha istikrarlı ve verimli öğrenme sağlamıştır. Bu yaklaşım, gerçek dünyadaki otonom sürüş senaryoları için sağlam DRL uygulamalarının geliştirilmesinde önemli ilerlemelerin potansiyelini vurgulamıştır.

J. X. Xu ve Yuan (2018), otonom sürüş için DDPG'yi geliştirmek üzere sezgisel başlatmayı kullanan bir yöntem olan HeuRL'yi tanıtmış ve standart DDPG'ye kıyasla TORCS simülatöründe dört kat daha hızlı öğrenme yakınsaması ve %90 daha az çarpışma elde etmiştir.

F. Yang ve diğ. (2018), otonom sürüş için DRL algoritmalarında kullanılan ödül fonksiyonunda yenilikçi geliştirmeler önermiştir. Araştırma direksiyon yumuşaklığı için bir ceza ekleyerek, TORCS ortamında hem DDPG hem de A3C algoritmalarını kullanarak direksiyon geçişlerinde gelişmiş kontrol göstermiştir. Bu

yaklaşım simüle edilmiş otonom sürüşün gerçekçiliğini ve güvenliğini önemli ölçüde geliştirerek gerçek dünya senaryolarında daha gelişmiş uygulamaların önünü açmayı hedeflemiştir.

Y. L. Chen ve diğ. (2019), yoğun trafikte şerit değiştirme davranışlarını geliştirmek için dikkat mekanizmalarına sahip hiyerarşik bir DRL algoritması sunmuş ve TORCS'da temel DRL yöntemlerine kıyasla gelişmiş güvenlik ve verimlilik göstermiştir.

Guckiran ve Bolat (2019), TORCS ortamında rekabetçi yarışlar için otonom ajanları eğitmek üzere SAC ve Rainbow DQN gibi gelişmiş DRL stratejilerini kullanmışlardır. Bu yöntemler, keşif ve sömürüyü etkili bir şekilde dengeleyerek tur sürelerinde ve yarış performansında önemli gelişmeler göstermiş ve bu algoritmaların yarış gibi daha karmaşık ve dinamik sürüş görevlerinde uygulanmasına yönelik öngörüler sunmuştur.

Z. Q. Huang ve diğ. (2019), DDPG kullanarak sürüş durumlarını eylemlerle doğrudan eşleştirerek otonom sürüş için yeni bir yaklaşım önermiştir. TORCS ortamında doğrulanan modelleri, verimli ve etkili sürüş politikalarını uçtan uca öğrenme becerisini göstermiş, gerçek dünyadaki otonom sürüş uygulamalarında derin pekiştirmeli öğrenmenin potansiyeli hakkında önemli bilgiler sunmuştur.

J. Chen ve diğ. (2020), insan geri bildirimini ödül yapısına dahil ederek sürüş davranışlarının kişiselleştirilmiş ve uyarlanabilir bir şekilde öğrenilmesini sağlayan yenilikçi bir DRL çerçevesi önermiştir. Sistem, dinamik olarak ayarlanmış bir ödül fonksiyonu ile geliştirilmiş DDPG algoritmasını kullanmaktadır; bu algoritma, yalnızca karmaşık ortamlara uyum sağlamada geleneksel DRL yaklaşımlarını geliştirmekle kalmamakta, aynı zamanda gerçek dünyadaki insan geri bildirimlerinden öğrenerek sürüş deneyimini kişiselleştirebilmektedir. Bu yöntem, genel otonom sürüş algoritmaları ile kişiselleştirilmiş kullanıcı deneyimleri arasındaki boşluğu doldurmayı hedeflemiştir.

Niu ve diğ. (2020), yüksek hızlı yarış görevlerinde güvenliği sağlamak ve politika performansını artırmak için DDPG'yi kural tabanlı ve veri güdümlü koruma

modülleriyle entegre eden iki aşamalı bir güvenli DRL yöntemi önermiş, TORCS'da sıfır güvenlik ihlali ve gelişmiş sürüş verimliliği elde etmiştir.

Stang ve diğ. (2020), otonom araçların yönlendirilmesinde DRL algoritmalarının etkinliğini araştırmıştır. SAC algoritmasına odaklanarak, diğer DRL yöntemlerine kıyasla öğrenme hızı ve yeni ortamlara uyum sağlama konusundaki üstün performansını ortaya koymuşlardır. Çalışma, girdinin simüle edilmiş ön kamera görüntüleri, çıktının ise araç kontrol eylemleri olduğu bir simülasyon düzeneği kullanarak SAC'nin gerçek dünyadaki otonom sürüş uygulamalarındaki potansiyelini göstermiştir.

Yuan ve diğ. (2020), otonom sürüşte kooperatif çarpışmadan kaçınma için Co-DDPG kullanan merkezi olmayan bir çerçeve olan RACE'i tanıtmış ve mevcut yöntemlere kıyasla TORCS simülatöründe çarpışma oranlarında önemli düşüşler ve gelişmiş sürüş verimliliği göstermiştir.

G. S. Zhu ve diğ. (2020), otonom araçların kontrol mekanizmalarını geliştirmek için DDPG'yi uygulamıştır. Bu çalışma, otonom araçların hassas manevraları için gerekli olan karmaşık, sürekli kontrol görevlerinin üstesinden gelmede Aktör-Kritik yöntemlerin potansiyelini vurgulamıştır. Ayrıca simüle edilmiş ortamlarda güvenlik ve verimlilikte önemli gelişmeler göstermiştir.

Agarwal ve diğ. (2021), CARLA simülatöründeki karmaşık kentsel sürüş senaryolarını ele almak için modüler ve öğrenme tabanlı yaklaşımları ustaca entegre eden bir DRL çerçevesi geliştirmiştir. Sofistike bir girdi temsili ve PPO algoritmasından yararlanan çerçeve, özellikle üst düzey navigasyon kararları ve trafik kurallarına uyulmasını gerektiren görevlerde iyileşme sağlayarak son teknoloji performansa ulaşmayı hedeflemiştir.

Ashraf ve diğ. (2021), DDPG'nin hiperparametrelerini optimize etmek için Balina Optimizasyon Algoritmasını kullanmış ve otonom sürüş görevleri için TORCS ortamında gelişmiş performans göstermiştir.

Cai ve diğ. (2021), 1/20 ölçekli RC-araba kullanarak hem simülasyon hem de gerçek dünya ortamlarında doğrulanan verimli ve güvenli otonom araba yarışı elde

etmek için Taklit Öğrenme ve Model tabanlı RL'yi entegre eden bir Derin Taklit Takviye Öğrenme çerçevesi geliştirmiştir.

Chang ve diğ. (2021), otonom sürüş kontrolünü geliştirmek için AirSim ortamında DDPG ve RDPG'yi kullanmış ve geleneksel kontrol stratejilerine kıyasla DRL yöntemleri aracılığıyla gelişmiş performans ve uyarlanabilirlik göstermiştir.

Y. Q. Wu ve diğ. (2021), RND merak mekanizması ve ikili eleştirmen ağını entegre eden gelişmiş bir PPO tabanlı yöntem önermiş ve TORCS ortamında şeritte tutma ve sollama görevlerinde gelişmiş eğitim verimliliği ve kontrol performansı göstermiştir.

R. Y. Yang ve diğ. (2021), TORCS kullanarak karmaşık senaryolarda otonom sürüş ajanlarını eğitmek için DDPG ve GAIL'i birleştiren yeni bir yaklaşım olan Columba'yı tanıtmışlardır. Columba, kendine özgü bir ödül mekanizmasının yanı sıra uzman ve anormal yörünge verilerinden yararlanarak, geleneksel yöntemlere kıyasla zorlu ortamlarda üstün performans göstermiştir.

Ahmed ve diğ. (2022), kentsel alanlarda otonom sürüşü geliştirmek için uygunluk öğrenimini DDPG ile entegre eden hibrit bir çerçeve önermiş, CARLA'da ve gerçek dünya video akışlarında kapsamlı simülasyonlarla doğrulanmış; çeşitli trafik koşullarında sağlam performans göstermiştir.

Anzalone ve diğ. (2022) CARLA'da otonom sürüş politikalarını eğitmek için PPO ile birleştirilmiş bir müfredat öğrenme yaklaşımı geliştirmiş, karmaşık davranışları aşamalı olarak öğrenmeye odaklanarak tüm şehirlerde tutarlı performans ve hava durumu değişikliklerine karşı kararlı olduğunu göstermiştir.

J. Y. Chen ve diğ. (2022) şehir içi otonom sürüşte yorumlanabilirliği ve performansı artırmak için sıralı gizli ortam modellemesini maksimum entropi RL ile birleştiren bir yöntem sunmuş ve CARLA'da temel yöntemlere göre önemli gelişmeler gösterdiği belirtilmiştir.

Hussonnois ve Jun (2022), CARLA ortamında Ape-X algoritmasını kullanarak uçtan uca bir otonom sürüş sistemi geliştirmiş ve dağıtılmış öncelikli deneyim

tekrarlama çerçevesinden yararlanarak diğer son teknoloji DRL tekniklerine kıyasla kentsel yol takip görevlerinde üstün performans göstermiştir.

Savari ve Choe (2022), CARLA simülatörünü kullanarak kesikli ve sürekli insan geri bildiriminin şerit tutma görevlerinde SAC'ın eğitim verimliliği üzerindeki etkisini araştırmıştır. Bulguları, ayrık geri bildirimin sürekli geri bildirime kıyasla eğitim verimliliğini ve performansını önemli ölçüde artırdığını ifade etmektedir.

Tian ve diğ. (2022), otonom sürüşte yol takibi için Davranış Klonlamayı DDPG ile entegre eden, öğrenme verimliliğini ve performansını artırmak için çift aktörlü bir ağ yapısı kullanan, Carsim/Simulink simülasyonları ve HIL sistem testleri aracılığıyla doğrulanan yeni bir çerçeve önermiştir.

Siboo ve diğ. (2023), TORCS simülatörünü kullanarak otonom sürüş görevlerinde DDPG ve PPO algoritmalarının karşılaştırmalı bir analizini yapmış; DDPG'nin özellikle çoklu ajan senaryolarında daha yüksek ödüller ve daha hızlı yakınsama açısından PPO'dan sürekli olarak daha iyi performans gösterdiği ifade edilmiştir.

Tammewar ve diğ. (2023), CarRacing-v2 ortamını kullanarak otonom yarış pisti navigasyonu bağlamında DQN, DDPG ve PPO algoritmalarının performansını analiz etmişlerdir. Çalışma, ε-bozunmalı DQN'nin üstün performans ve kararlılık sağladığını ortaya koymuştur.

Jinkyu Kim ve diğ. (2018), sürücüsüz araçlar için metinsel açıklamalar üretmeye yönelik yeni bir yaklaşım sunarak, araç kontrolörünün iç durumuna dayanan iç gözlemsel açıklamalara odaklanmıştır. Yöntem, araç komutlarını etkileyen görüntü bölgelerini belirlemek için görsel bir dikkat modeli kullanıyor ve bunları açıklamalar üretmek için dikkat tabanlı bir videodan metne modeliyle eşlemiştir. Yaklaşım, BDD-X veri kümesi kullanılarak doğrulanmış ve araç eylemlerinin insan benzeri rasyonalizasyonlarını üretmede dikkat haritalarının avantajını göstermiştir.

J. Kim ve Canny (2017), araç kontrolü için kullanılan DNN'lerin karar verme sürecini anlamaya yardımcı olan nedensel dikkati görselleştirerek, sürücüsüz araçlarda yorumlanabilir öğrenme için bir model sunmuştur. Model, görsel özellikleri kodlamak için bir CNN ve ardından giriş görüntüsündeki etkili bölgeleri vurgulamak için bir

görsel dikkat mekanizması kullanmaktadır. Yöntem, üretilen dikkat haritalarının kontrol doğruluğundan ödün vermediğini ve ağın davranışının net görsel açıklamalarını sağladığını gösteren kapsamlı sürüş veri kümeleri üzerinde doğrulanmıştır. Bu yaklaşım, karar verme süreçlerini kullanıcılar ve paydaşlar için daha şeffaf ve yorumlanabilir hale getirerek sürücüsüz araç sistemlerinin açıklanabilirliğini artırmayı hedeflemektedir.

Bojarski ve diğ. (2018), özellikle gerçek zamanlı otonom sürüş uygulamaları için uyarlanmış CNN tabanlı sistemler için verimli bir görselleştirme yöntemi olan VisualBackProp'u tanıtmıştır. VisualBackProp, değerleri ağ katmanları boyunca geri yayarak yüksek çözünürlüklü görselleştirmeler üretmiş ve giriş görüntüsünde ağın çıktısını etkileyen ilgili bölgeleri belirginleştirmiştir. Yöntem, NVIDIA'nın PilotNet'ine entegre edilmiş, sürücüsüz araçların direksiyon kararlarını etkileyen şerit işaretleri ve araçlar gibi önemli görsel ipuçlarını vurgulama kabiliyeti gösterilmiştir. VisualBackProp'un, benzer görselleştirme kalitesini korurken Katman Bazlı Alaka Yayılımından (LRP) önemli ölçüde daha hızlı olduğu gösterilmiştir, bu da gerçek zamanlı otonom sürüş bağlamlarında CNN'lerde hata ayıklama ve yorumlama için verimli kılmıştır.

Mori ve diğ. (2019), aracın kontrol kararları için görsel açıklamalar sağlamak üzere bir Dikkat Dal Ağı (ABN) entegre eden uçtan uca öğrenme tabanlı otonom sürüş için bir yöntem sunmuştur. Model, araç hızını ek bir girdi olarak dahil ederek ve Ağırlıklı Küresel Havuzlama (WGP) katmanı kullanarak hem direksiyon hem de gaz kelebeği kontrolünü geliştirmektedir. ABN çerçevesi, ağın kararlarını etkileyen bölgeleri vurgulayan dikkat haritaları oluşturarak kendi kendine sürüş sisteminin yorumlanabilirliğini arttırmıştır. Yöntemin etkinliği GTAV tabanlı bir sürüş simülatörü kullanılarak doğrulanmış, istikrarlı kontrol ve ağ çıktılarının net görsel açıklamaları gösterilmiştir.

Rahimpour ve diğ. (2019), otonom sürüş için tasarlanmış, bağlama duyarlı bir yol kullanıcısı önem tahmin sistemi olan iCARE (context-aware road-user importance estimation system) modelini önermiştir. Model, yol kullanıcılarının (trafikteki diğer sürücüler) önemini değerlendirmek için hem yerel özelliklerden (görünüm ve konum) hem de küresel bağlamdan (ego aracının gelecekteki yolu) yararlanmaktadır. Kentsel kavşaklardan oluşan bir veri kümesi üzerinde değerlendirilen iCARE modeli, kapsamlı

olay algılama ve niyete dayalı bağlamı etkili bir şekilde entegre ederek, temel yöntemlere göre üstün performans göstermiştir. Bu yaklaşım, kritik yol kullanıcılarını belirleyerek ve önceliklendirerek sürücüsüz araçlarda karar verme şeffaflığını ve güvenliği arttırmayı hedeflemiştir.

Cultrera ve diğ. (2020), model yorumlanabilirliğini ve performansını artırmak için görsel bir dikkat mekanizması içeren otonom sürüş için uçtan uca bir öğrenme çerçevesi sunmuştur. Yöntem, RGB görüntülerini üst düzey komutlara dayalı direksiyon açılarıyla eşleştirmek için koşullu taklit öğrenimini kullanır. Görsel dikkat mekanizması, görüntülerde sürüş kararlarını etkileyen önemli bölgeleri tanımlayarak yorumlanabilir görsel açıklamalar sağlamayı hedefler. CARLA sürüş simülatörü kullanılarak değerlendirilen yaklaşım, gelişmiş performans ve kritik görsel ipuçlarını vurgulama becerisi göstermekte, böylece karar verme sürecini daha şeffaf ve anlaşılır hale getirmektedir.

Y. Xu ve diğ. (2020), yorumlanabilirliği ve performansı artırmak için eyleme neden olan nesnelere odaklanarak uçtan uca ve boru hattı yöntemlerini entegre eden otonom sürüşe hibrit bir yaklaşım getirmiştir. Yöntem hem sürüş eylemlerini hem de bunlara karşılık gelen açıklamaları tahmin eden ve sistemin karar verme sürecinin anlaşılmasını geliştiren çok işlevli bir CNN mimarisi kullanmaktadır. BDD-OIA veri setini kullanan model, nesne tespiti için FasterR-CNN ve eyleme neden olan nesneleri tanımlamak için global bir olay bağlam modülünü birleştirmiştir. Çalışma, açıklamaların dahil edilmesinin yalnızca eylem tahminini iyileştirmekle kalmayıp aynı zamanda daha net nedensel ilişkiler sağladığını ve karmaşık sürüş ortamlarında gelişmiş sonuçlar elde ettiğini göstermektedir.

Schneider ve diğ. (2021); sistem şeffaflığının kullanıcı deneyimi, otonom sürüşte güvenlik, kontrol algıları üzerindeki etkilerini araştırmıştır. Çalışma, otonom sürüş sırasında yapılan canlı açıklamaların yolcuların anlayışını, kullanım kolaylığını ve algılanan kontrolü önemli ölçüde artırdığını göstermiştir. Hem canlı hem de geriye dönük açıklamalar olumsuz kullanıcı deneyimini azaltabileceği belirtilmiştir. Bu araştırma, otonom sürüş sistemlerinde kullanıcı kabulünü ve deneyimini iyileştirmede şeffaflığın ve gerçek zamanlı açıklamaların önemini vurgulamıştır.

Mankodiya ve diğ. (2021), güven ve güvenilirliği artırmak için XAI'nın otonom araç sistemlerine entegrasyonunu araştırıyor. Çalışma, karar verme sürecinin opak olmadığı ve kullanıcılar tarafından kolayca anlaşılamadığı, kara kutu olarak hareket eden YZ modellerinin zorluklarını vurgulamıştır. Bu doğrultuda XAI yeteneklerine sahip karar ağacı tabanlı bir model uygulayarak, "Vehicular Adhoc Ağları" (VANETs) içinde karar verme süreçlerinde şeffaflık sağlamayı hedeflemişlerdir. Bu yaklaşım sadece hatalı araçların tespit edilmesine yardımcı olmakla kalmamış, aynı zamanda YZ'nin muhakemesini kullanıcılar için erişilebilir ve anlaşılabilir hale getirerek AV sistemlerine olan genel güveni de arttırmayı amaçlamıştır.

Shen ve diğ. (2022), otonom araçlar için açıklamaların gerekliliği üzerine kapsamlı bir çalışma yürütmüş ve farklı sürüş senaryoları ile sürücü tiplerinin açıklama ihtiyacını nasıl etkilediğine odaklanmıştır. Bulgular, otonom araçlara güven oluşturmada bağlama duyarlı açıklamaların önemini vurgulamakta ve temkinli sürücülerin agresif sürücülere kıyasla daha fazla açıklama talep ettiğini göstermiştir. Bu araştırma otonom sürüşte XAI'nın geliştirilmesini desteklemek için değerli öngörüler ve veri seti sağlamıştır.

Dong ve diğ. (2023) tarafından yapılan araştırmada, otonom sürüş sistemlerinde kullanıcı güvenini artırmada XAI'nın önemi vurgulanmıştır. Çalışmada otonom araçlar tarafından alınan kararları açıklamak için hem görsel girdileri hem de dil açıklamalarını kullanan yenilikçi, çok modlu derin öğrenme mimarisi tanıtılmıştır. Bu yaklaşım sadece yapay zekanın karar verme sürecinin daha iyi anlaşılmasını sağlamakla kalmamış, aynı zamanda otonom sürüş sistemlerinin güvenilirliğini ve şeffaflığını da önemli ölçüde arttırmıştır. Karar verme sürecini görüntü tabanlı bir dil oluşturma görevi olarak ele alan metodolojileri, XAI'nin otonom sürüş için DRL çerçevelerine nasıl entegre edilebileceğine dair değerli bilgiler sunmuş; böylece YZ kararlarının son kullanıcılar için yorumlanabilir ve gerekçelendirilebilir olmasını sağlama konusunda literatürdeki kritik bir boşluğu ele almıştır.

Atakishiyev ve diğ. (2023), otonom sürüş sistemlerinin güvenliğini ve güvenilirliğini artırmak amacıyla otonom kontrol, XAI ve mevzuata uygunluğu entegre etmek için sağlam bir çerçeve önermiştir. Yaklaşımları, bu tür sistemlerin yalnızca etkili değil, aynı zamanda düzenleyici otoriteler ve genel halk için

anlaşılabilir ve güvenilir olmasını sağlamak için otonom sürüş ortamında XAI'ye duyulan ihtiyacı vurgulamıştır. Önerilen çerçeve, operasyonel kararlarını şeffaf ve açıklanabilir hale getirerek otonom araçların benimsenmesini kolaylaştırmayı amaçlamaktadır.

Nazat ve diğ. (2024), otonom sürüş sistemlerinde anomali tespitinin kritik zorluklarını ele alırken, sistem şeffaflığını ve kullanıcı güvenini önemli ölçüde artıran yenilikçi bir XAI çerçevesi önermiştir. Çalışmaları, yeni özellik seçim tekniklerini tanıtmakta ve VANET'ler içindeki YZ modellerinin karar verme süreçlerini aydınlatmak için XAI yöntemlerini benimsemektedir. Bu yaklaşım sadece YZ güdümlü anomali tespitinin yorumlanabilirliğini iyileştirmekle kalmamakta; aynı zamanda YZ kararları için açık, anlaşılır açıklamalar sağlayarak otonom araçların güvenilirliğini de desteklemektedir.

Atakishiyev ve diğ. (2024), VANET'lerde anomali tespiti için kullanılan YZ modellerinin yorumlanabilirliğini ve güvenilirliğini artırmak için tasarlanmış yeni bir XAI çerçevesi sunmuşlardır. Popüler SHAP yönteminden türetilen iki yenilikçi özellik seçim tekniği sunarak, mevcut YZ modellerinde opak karar verme süreçlerinin kritik zorluğu ele alınmıştır. Bu tekniklerin verimliliği; geleneksel özellik seçim yöntemlerine kıyasla üstün performans göstermiş, iki gerçek dünya otonom sürüş veri kümesi üzerinde yapılan kapsamlı değerlendirmelerle doğrulanmıştır. Bu çalışma otonom araçların güvenliğini sağlamada XAI uygulamasını ilerleterek, kritik uygulamalarda YZ sistemlerinin şeffaflığını ve güvenilirliğini artırmayı amaçlayan gelecekteki araştırmalar için bir temel oluşturmaktadır.

Abu Dabi Otonom Yarış Ligi (A2RL 2024), ASPIRE tarafından düzenlenen ve Yas Marina Pisti'nde otonom araçların sergilendiği öncü bir motor sporları etkinliğidir. Dallara yapımı Super Formula SF23 araçlarının yer aldığı lig, yüksek hızlı ve kontrollü bir ortamda otonom sürüş teknolojisinin sınırlarını zorlamıştır. Etkinlik, otonom araçların yüksek hızlı yarışlardaki önemli potansiyelini vurgulamış ve yapay zekanın yeteneklerini ve gelecekteki potansiyelini sergilemiştir.

Bu alt bölümde incelenen çalışmalar, derin pekiştirmeli öğrenmenin (DRL) otonom sürüş uygulamalarında sağladığı ilerlemeleri ve zorlukları kapsamlı bir şekilde ortaya koymaktadır. Farklı DRL algoritmalarının performansını ve uygulanabilirliğini

değerlendiren bu çalışmalar, otonom sürüş sistemlerinin daha güvenli, verimli ve kullanıcı dostu hale getirilmesinde önemli katkılar sunması hedeflenmiştir. Özellikle XAI yöntemlerinin entegrasyonu, sistemlerin şeffaflığını artırarak kullanıcıların ve düzenleyici kurumların/organizasyonların güvenini kazanma potansiyeli taşımaktadır. Bununla birlikte mevcut çalışmalar, eğitim verisi gereksinimleri, karar verme süreçlerinin şeffaflığı ve algoritmaların adaptasyonu gibi konularda hala iyileştirilmesi gereken alanlar olduğunu göstermektedir. Bu tez kapsamında geliştirilen XAI yöntemleri ile DRL algoritmalarının otonom sürüşe uygulanmasında belirtilen sorunlara çözüm üretilmesi ve bu alanda yeni katkılar sağlanması hedeflenmektedir.

Tablo 4.1 incelendiğinde en çok tercih edilen DRL algoritmalarından olan TD3, DDPG ve PPO, bu çalışma için seçilmiştir. Simülasyon ortamı olarak da benzer şekilde TORCS tercih edilmiştir. Bu bölümde ödül fonksiyonu ve parametreler bu çalışmalara göre ayarlanmıştır. TD3, DDPG'nin geliştirilmiş bir versiyonu olup, fazla yakınsama önyargısını azaltmak için iki kritik ağı kullanır ve sürekli eylem alanlarında daha stabil ve güvenilir sonuçlar elde edilmesini sağlar. DDPG, sürekli aksiyon alanlarına sahip ortamlarda başarılı sonuçlar veren ve politika dışı öğrenme sağlayan bir algoritma olarak öne çıkar; özellikle otonom sürüş gibi yüksek boyutlu aksiyon alanlarında etkili bir şekilde çalışabilmesi nedeniyle seçilmiştir. PPO ise basitliği, yüksek performansı ve gradyan tabanlı politika yöntemlerinde yaygın olan politika güncelleme sorunlarını güvenli bir güncelleme mekanizması ile çözer. Bu üç algoritmanın kombinasyonu, otonom sürüş uygulamalarında geniş kapsamlı ve güvenilir bir performans analizi yapabilmek için ideal bir temel sağlaması hedeflenmiştir.

SHAP modeli, DL modellerinin kararlarını açıklamak için güçlü ve etkili bir yöntem olarak tercih edilmiştir. SHAP, her bir özelliğin model çıktısına olan katkısını anlamaya olanak tanıyarak, otonom sürüş modellerinin kararlarını açıklamak için kritik bir araç sunmaktadır. Bu sayede model kararlarının şeffaf bir şekilde açıklanması sağlanarak, kullanıcıların ve düzenleyici organizasyonların sisteme olan güveni artırılması hedeflenir. Özellikle GDPR gibi yasal düzenlemeler, DL modellerinin kararlarını açıklayabilir olmasını gerektirdiğinden, SHAP'ın bu yasal gerekliliklere uygun çözümler sunabileceği amaçlanmıştır. Bu nedenlerden dolayı SHAP modeli,

otonom sürüş sistemlerinin açıklanabilirliğini ve güvenilirliğini artırmak için tercih edilmiştir.

Tablo 4.1: DRL algoritmalarıyla uçtan uca otonom sürüş çalışmaları.

Çalışma	Algoritma	Simülasyon Ortamı	Ödül Fonksiyonu	Durum (S)	Eylem (A)
Xia ve diğ. (2016)	DQL	TORCS	mesafe tabanlı	TORCS	ABS
Gu ve diğ. (2017)	DDPG	TORCS	açı ve pist üstü pozisyon	TORCS	S
Ye ve diğ. (2017)	DDPG	TORCS	negatif kaçınma fonksiyonu	TORCS	ABS
Zuo ve diğ. (2017)	DDPG	TORCS	hız ve açı	TORCS	ABS
Chishti ve diğ. (2018)	DQN & CNN	Ö.T.S.O.	trafik işaretlerine ve yol koşullarına uyum	Kamera	ABS
Fayjie ve diğ. (2018)	DQN	Ö.T.S.O. (Unity)	güvenli sürüş tabanlı	Kamera+LiDAR	RLF
Hilleli ve El-Yaniv (2018)	DDQN	Assetto Corsa	İnsan sürücü ve denetimli öğrenme	Kamera	S
Kaushik ve diğ. (2018)	DDPG	TORCS	şerit koruma ve sollama	TORCS	ABS
Liang ve diğ. (2018)	DDPG & CIRL	CARLA	eylemler, hız kontrolü ve kazalar	Kamera ve diğer sensörler	ABS & F
K. Liu ve diğ. (2018)	DDPG	TORCS	şerit koruma, hız, açı	TORCS	ABS
Luo ve diğ. (2018)	OPGD	TORCS	-	TORCS	ABS
J. X. Xu ve Yuan (2018)	DDPG	TORCS	şerit koruma ve hız	TORCS	ABS
F. Yang ve diğ. (2018)	DDPG & A3C	TORCS	direksiyon çevirme hızının yumuşaklığı	TORCS	S
Y. L. Chen ve diğ. (2019)	DDPG	TORCS	şerit koruma, hız, rakibi sollama ve ihlal cezaları	TORCS, Kamera	ABS
Guckiran ve Bolat (2019)	SAC & Rainbow DQN	TORCS	şerit koruma, hız, açı	TORCS	ABS
Z. Q. Huang ve diğ. (2019)	DDPG	TORCS	şerit koruma, hız, açı	TORCS	ABS
D. Li ve diğ. (2019)	DPG	TORCS	pozisyon ve hatalar için ceza	TORCS	ABS
J. Chen ve diğ. (2020)	DDPG PORF	CARLA & G.D.	PORF	Kamera ve diğer sensörler	ABS
S. Y. Chen ve diğ. (2020)	DDPG	TORCS & Gazebo	şerit koruma, hız, açı ve kaza durumunda ceza	Kamera, Lidar ve diğer sensörler	ABS
Niu ve diğ. (2020)	DDPG	TORCS	adaptif güvenlik, verim ve stabilite	TORCS	ABS
M. X. Peng ve diğ. (2020)	DDPG & IL	CARLA	hız ve açı	Kamera ve diğer sensörler	RLF
Stang ve diğ. (2020)	SAC	AirSim	hız ve şerit takibi	Kamera	ABS
Yuan ve diğ. (2020)	DDPG	TORCS	şerit koruma, süre, kaza ve ihlalden kaçış	TORCS	ABS

Tablo 4.1 DRL algoritmalarıyla uçtan uca otonom sürüş çalışmaları(devamı).

G. S. Zhu ve diğ. (2020)	**DDPG**	**TORCS**	**şerit koruma, hız, kaza durumunda ceza**	**TORCS**	**ABS**
Zou ve diğ. (2020)	DDPG	TORCS	şerit koruma, hız, kaza durumunda ceza	TORCS & Kamera	ABS
Agarwal ve diğ. (2021)	PPO	CARLA	şerit koruma, hız, hata ve ihlallerde ceza	Kuşbakışı görüntü ve diğer sensör verileri	ABS
Ahmed ve diğ. (2021)	DQN	CARLA	şerit koruma, açı	Kamera ve diğer sensörler	ABS
Alighanbari ve Azad (2021)	DDPG	SUMO	eylemler, hız kontrolü ve kazalar	trafik ve tüm araçların bilgisi	AB
Anzalone ve diğ. (2021)	PPO	CARLA	hız, açı ve ihlal durumunda cezalar	Kamera, LiDAR ve diğer sensörler	(AB)S
Ashraf ve diğ. (2021)	DDPG	TORCS	hız, açı ve ihlal durumunda cezalar	TORCS	ABS
Cai ve diğ. (2021)	Model Tabanlı RL & IL	G.D.	hızlı ve güvenli sürüş	Kamera, IMU	AS
Chang ve diğ. (2021)	DDPG, RDPG	AirSim	şerit koruma, stabilite, ihlal durumunda ceza	Kamera, IMU	ABS
Chukamphaeng ve diğ. (2021)	TD3, SAC	Ö.T.S.O. (PyBullet)	yön odaklı ödül ve ihlal durumunda ceza	Kamera, mesafe sensörü	S
C. X. Huang ve diğ. (2021)	DDPG	CARLA	hız, aracın yol üzerindeki durumu, kaza durumunda ceza	Kamera, IMU	ABS
Jin ve diğ. (2021)	DDPG, MAPDDPG	TORCS	şerit koruma, hız, kaza durumunda ceza	TORCS & Kamera	ABS
Kishikawa ve Arai (2021)	TD3	TORCS	şerit koruma, hız, hedef, kaza durumunda ceza	TORCS	S
K. Lee ve diğ. (2021)	AIRL & IL	TORCS & Gazebo & G.D.	hız ödülü ve ihlal durumunda ceza	TORCS & Kamera	AS
B. Y. Peng ve diğ. (2021)	DDDQN	TORCS	Şerit koruma, ihlal halinde ceza	TORCS & Kamera	S
Perez-Gill ve diğ. (2021)	DDPG	CARLA	Şerit koruma, hız, açı	Hedef konumu, hız, şerit bilgisi ve açı	AS
Quek ve diğ. (2021)	DQN, DDQN	Ö.T.S.O. (Py, Unity)	hız ve cezalar	Ortam resmi	AS
T. H. Wang ve diğ. (2021)	POAC	TORCS	yakıt verimliliği, şerit konumu, hız limiti	TORCS, Kamera	AS
Z. L. Wang ve diğ. (2021)	SAC	CARLA	hız, açı ve ihlal durumunda cezalar	Kamera, LiDAR	ABS

Tablo 4.1 DRL algoritmalarıyla uçtan uca otonom sürüş çalışmaları(devamı).

Y. Q. Wu ve diğ. (2021)	**PPO, TD3, DDPG**	**TORCS**	**hız, açı ve ihlal durumunda cezalar**	**TORCS**	**ABS**
R. Y. Yang ve diğ. (2021)	DDPG, GAIL	TORCS	Columba	TORCS	ABS
Zou ve diğ. (2021)	DDPG	TORCS	-	TORCS & Kamera	ABS
Ahmed ve diğ. (2022)	DDPG	CARLA	Şerit koruma, hız, kaza durumunda ceza	Trafik bilgileri, mesafe ve diğer sensörler	ABS
Anzalone ve diğ. (2022)	PPO, CIRL	CARLA	Planlanan rota, hız, kaza durumunda ceza	Kamera, araç ve yol durumu, navigasyon	ABS
J. Y. Chen ve diğ. (2022)	DQN, DDPG, SAC, TD3	CARLA	şerit koruma, hız, sürüş yumuşaklığı, kaza durumunda ceza	Kamera, LiDAR	ABS
Hussonnois ve Jun (2022)	Ape-X	CARLA	hedefe varış için ödül ve kaza durumunda ceza	Kamera ve diğer sensörler	ABS
D. F. Li ve diğ. (2022)	PPO	CARLA	-	Görüntü	-
Lin ve diğ. (2022)	PPO (APF-DPPO)	Ö.T.S.O. (Unity)	APF	Kuşbakışı görüntü ve diğer sensör verileri	AS
H. C. Liu ve diğ. (2022)	SAC	CARLA	güvenli ve verimli sürüş tabanlı	Kuşbakışı görüntü ve araç durumu	AB
Riboni ve diğ. (2022)	DDQN, DDDQN	-	güvenli sürüş tabanlı	Görüntü	AS
Savari ve Choe (2022)	SAC	CARLA	hız ve ivmelenme, şerit ihlali cezası	Kamera, araç durumu	AS
Tian ve diğ. (2022)	DDPG	CarSim/Simulink	hız, açı ve izleme hassasiyeti	Ortamdan ölçülebilir sensör bilgileri	AS
Siboo ve diğ. (2023)	DDPG, PPO	TORCS	hız, açı, kaza/ihlal durumunda ceza	TORCS	ABS
Hossain (2023)	DQN	CARLA		Araç ve trafik durumu	RL, Hız
Z. Huang ve diğ. (2023)	SAC	SMARTS	Kaza, hedef	kuşbakışı görüntü	RL, Hız
Tammewar ve diğ. (2023)	DQN, DDPG, PPO	CarRacing-v2	Şerit koruma, ihlal halinde ceza	Görüntü	ABS
Ashwin ve Naveen Raj (2023)	DDPG	TORCS	Şerit koruma, hız, sollama, kaza durumunda ceza	TORCS	ABS

A: Gaz/ivmelenme, B: Fren, S: Direksiyon açısı, RL: Sağ Sol F: İleri, Ö.T.S.O: Özel tasarlanmış simülasyon ortamı, G.D: Gerçek dünya. Durum sütunundaki TORCS simülatörün içerisindeki sensör verilerinin kullanıldığını belirtir.

4.3 Simülasyon Ortamı

Bu tez çalışmada otonom sürüş algoritmalarının eğitimi ve testi için TORCS (The Open Racing Car Simulator) kullanılmıştır. Bu alt bölümde TORCS hakkında detaylı bilgiler işlenmiştir. Ayrıca TORCS'un kurulumu, yapılandırılması ve simülasyon parametrelerinin ayarlanması konuları ele alınmıştır.

4.3.1 TORCS

Simülasyon ortamının seçimi, DRL algoritmalarının gerçek dünya senaryolarına benzer koşullarda test edilmesi ve performanslarının değerlendirilmesi açısından kritik öneme sahiptir. TORCS'un tercih edilme sebepleri arasında gerçekçi araç dinamikleri ve yarış senaryoları, ihtiyaca göre özelleştirme imkânı, düşük sistem gereksinimleri yer almaktadır.

TORCS C++ ile yazılmış açık kaynaklı bir 3-boyutlu araba yarışı simülatörüdür. Otonom sürüş sistemleri ve yapay zekâ algoritmaları için geliştirilmiş geniş bir platform sağlar. Windows, MacOS ve GNU/Linux gibi çeşitli işletim sistemlerinde çalışabilen TORCS; gerçekçi bir araç dinamiği modellemesi, çeşitli araç ve pist seçenekleri ile kullanıcıların algoritmalarını gerçek dünya şartlarına benzer bir ortamda test etmelerine imkân sağlamaktadır. Düşük performans gereksinimleri ile daha erişilebilir bir test ortamı sunmakta ve araştırmacılara sürüş davranışı simülasyonları üzerinde tam kontrol imkânı verir.

TORCS, araştırmacılara sürüş davranışı simülasyonları üzerinde tam kontrol imkânı vererek karmaşık sürüş senaryolarını modelleme fırsatı sunar. Simülasyon içerisinde araçların çevresel şartları algılaması için çeşitli sanal sensörler yer almaktadır. Aracın şerit içerisindeki pozisyonu, şeritlere olan uzaklık bilgileri, hız ve ivme ölçerlerden gelen bilgiler, kamera sensörleri bunlardan bazılarıdır. Bu sensör bilgileri sürüş algoritmalarında büyük rol oynamaktadır. Kullanıcılar yerel ağ üzerinden simülatöre bağlanabilmekte ve bu sayede kolaylıkla çeşitli kodların içerisine entegre edilebilmektedir. Böylece kullanıcı ya da otonom sürüş algoritması, simülasyona doğrudan erişebilir ve gerçek zamanlı olarak kontrol edebilir.

TORCS, açık kaynaklı olması ve içerdiği ayarların çeşitliliği ile kullanıcıların kendi ihtiyaçlarına göre özelleştirmelerine imkân tanır. Kurulum süreci basit adımlardan oluşur ve kullanıcılar simülasyonu kendi sistemlerine kolayca entegre edebilirler. Bu bağlamda tez kapsamında çeşitli ihtiyaçlara göre özelleştirmeler yapılmıştır. TORCS'da simülasyon parametrelerini de kullanıcıların ihtiyaçlarına göre ayarlamak mümkündür. Bu parametreler arasında araçların maksimum hızları, yarış koşulları, hava durumu ve görüş mesafesi gibi değişkenler yer alır. Kullanıcılar bu parametrelerle farklı trafik ve yol koşullarını simüle edebilirler. Örneğin, yağmurlu veya sisli bir hava durumunda aracın yol tutuşunun nasıl etkileneceğini gözlemlemek mümkündür. Ayrıca kullanıcılar simülasyon sırasında çeşitli senaryolar oluşturarak algoritmalarının farklı durumlara adaptasyonunu test edebilirler.

TORCS 19 farklı tipte sensör bilgisi sağlar. Bu sensörler ve bu sensörlere ait bilgiler Tablo 4.2'de verilmiştir. Sensörlerin çıktıları daha sonra normalize edilerek yapay sinir ağlarına girdi olarak verilmiştir. Ayrıca simülatörde aracı kontrol edebilmek için sağlanabilecek girdiler Tablo 4.3'te gösterilmiştir.

Tablo 4.2: Simülatörden alınan sensör verileri ve özellikleri (Loiacono ve diğ. 2013).

Özellik	Aralık	Açıklama
Açı (angle)	[-π, +π] (rad)	Aracın yönü ile pist ekseninin açısı.
Mevcut Tur Zamanı (curLapTime)	[0, +∞) (s)	Mevcut turda geçen süre.
Hasar (damage)	[0, +∞) (puan)	Aracın mevcut hasar durumu; değer yüksekse hasar da yüksek.
Başlangıçtan Uzaklık (distFromStart)	[0, +∞) (m)	Start çizgisinden itibaren aracın aldığı mesafe.
Kat Edilen Mesafe (distRaced)	[0, +∞) (m)	Yarış başlangıcından itibaren aracın kat ettiği mesafe.
Yakıt (fuel)	[0, +∞) (l)	Aracın mevcut yakıt seviyesi.
Vites (gear)	{-1,0,1,...,6}	Mevcut vites; -1 geri, 0 boş.
Son Tur Zamanı (lastLapTime)	[0, +∞) (s)	Son tamamlanan tur için süre.
Rakipler (opponents)	[0,200] (m)	36 sensör, her biri 200m içindeki en yakın rakibi algılar.
Yarış Pozisyonu (racePos)	{1,2,...,N}	Diğer araçlara göre yarış içindeki pozisyon.
Motor Devri (rpm)	[0, +∞) (rpm)	Motorun dakikadaki devir sayısı.
HızX (speedX)	(−∞,+∞) (km/s)	Aracın uzunlamasına hızı.
HızY (speedY)	(−∞,+∞) (km/s)	Aracın enlemesine hızı.
HızZ (speedZ)	(−∞,+∞) (km/s)	Aracın dikey hızı.
Pist Pozisyonu (trackPos)	(−∞,+∞)	Pist eksenine göre aracın pozisyonu; 0 ise eksen üzerinde.
Tekerlek Dönüş Hızı (wheelSpinVel)	[0, +∞] (rad/s)	Tekerleklerin dönüş hızlarını gösteren 4 sensör.
Z Yüksekliği (z)	[−∞, +∞] (m)	Aracın pist yüzeyinden Z eksenindeki yüksekliği.
Pist (track)	[0,200] (m)	Pist limitlerine kadar olan mesafeyi gösteren 19 sensör. Gürültülü ortamlarda %10 standart sapma ile ölçüm yapar.
Odak (focus)	[0,200] (m)	Belirli bir yönde 200 metreye kadar alanı tarayan 5 odak sensörü. Sensörler, gürültü seçeneği aktifken %1 standart sapma ile çalışır.

Tablo 4.3: Simülatörün eylem uzayı ve kontrol parametreleri (Loiacono ve diğ. 2013)

Özellik	Aralık	Açıklama
Gaz (acceleration)	[0,1]	Sanal gaz pedalı; 0 gaz yok, 1 tam gaz.
Fren (brake)	[0,1]	Sanal fren pedalı; 0 fren yok, 1 tam fren.
Debriyaj (clutch)	[0,1]	Sanal debriyaj pedalı; 0 debriyaj yok, 1 tam debriyaj.
Vites (gear)	-1,0,1,...,6	Vites değeri; -1 geri vites, 0 boş vites.
Direksiyon (steering)	[-1,1]	Direksiyon değeri: -1 tam sağa, +1 tam sola dönüş, bu 0.366519 rad açıya karşılık gelir.
Odak (focus)	[-90,90]	Odak yönü; derece cinsinden.
Meta (meta)	0,1	Meta-kontrol komutu: 0 hiçbir işlem yapma, 1 yarışı yeniden başlatma talebi.

TORCS birçok farklı mod içerir. Bunlardan bazıları eğitim modu ve hızlı yarış modudur. Eğitim modunda tek bir araç turlayabilirken yarış modunda birden fazla araç yarışabilmektedir. Yarış modlarında oyuncu, oyunun kendi içerisindeki otomatik

yarışan botlar, dışarıdan bağlanan otonom sürüş modelleri aynı anda yarışabilmektedir.

Eğitim esnasında ve yarış esnasında kullanılabilecek bazı komutlar aşağıda verilmiştir. Bu komutlar kullanıcıya esneklik sağlayarak eğitim sürecini kolaylaştırmaktadır. Simülatörü başlatmak için GNU/Linux işletim sisteminde uçbirime "`torcs`" komutu girilmesi yeterlidir. Ek olarak girilebilecek bayraklar/komutlar:

- `-nofuel`: yakıt limitini kaldırır.
- `-nodamage`: aracın hasar almasını kapatır.
- `-nolaptime`: tur süresi sınırını kaldırarak eğitim sürecinde aracın turunu tur uzun sürdüğü için kesmez.
- `-vision`: eğer kamera sensörü bilgisi isteniyorsa girilmesi gerekir.
- `-j double`: `j` bayrağı simülasyonun zaman oranını belirler. Eğer double kısmına girdi olarak 1 değeri girilirse $1/double$ olarak hesaplandığında 1 değeri elde edilir. Yani simülasyon gerçek zamanlı olarak çalışır. Ancak 0.25 (minimum girilebilecek değer) değeri girilirse simülasyon 4 kat hızlanır. Yani simülasyondaki 1 saniye gerçek dünyada 0.25 saniyeye eşittir.

Aşağıda çalışma için TORCS kurulumunun adımlarına sırasıyla değinilmiştir. Ubuntu 20.04 işletim sistemi üzerinde TORCS kurulumu şu adımlarla yapılabilir:

- TORCS kaynak paketi resmî sitesinden indirilir.
- Kurulum öncesinde gerekli paketleri yüklemesi aşağıdaki komutlar ile yapılır.

Bunların haricinde PLIB kurulumu da yapılması gerekir.

```
    sudo apt-get install libglib2.0-dev libgl1-mesa-dev
libglu1-mesa-dev freeglut3-dev libplib-dev libopenal-dev
libalut-dev libxi-dev libxmu-dev libxrender-dev libxrandr-dev
libpng12-dev
    sudo apt install ffmpeg
    sudo apt-get update
    sudo apt-get install libxxf86vm-dev
```

- Sıkıştırılmış `torcs-1.3.4.tar` isimli dosya "`tar xfvj torcs-1.*.*.tar*`" komutu ile çıkartılır.
- Ardından aşağıdaki komutlar ile kurulum yapılabilir.

```
./configure
make
make install
make datainstall
```

- Bir diğer alternatif olarak Visual TORCS versiyonu da tercih edilebilir. Bu versiyon görsel derin pekiştirmeli öğrenme yöntemleri için modifiye edilmiştir.
- İlk olarak "`git clone https://github.com/ugo-nama-kun/gym_torcs.git`" komutu ile github üzerinden dosya çekilir. Yukarıdaki gerekli paketler yine benzer şekilde kurulur.
- Ardından Ubuntu 20.04 GCC versiyonundaki hatayı almamak için aşağıdaki işlemler gerçekleştirilir. İlk olarak "geometry.cpp" dosyasının bulunduğu dizine geçilir. Dosya açılır ve `isnan` kod diziminin olduğu satırda `std::isnan` şeklinde düzenleme yapılır.

```
cd gym_torcs/vtorcs-RL-color/src/drivers/olethros/
nano ./geometry.cpp
isnan → std::isnan
```

- Bu düzenlemeler yapıldıktan sonra aşağıdaki komutlar ile kurulum gerçekleştirilebilir.

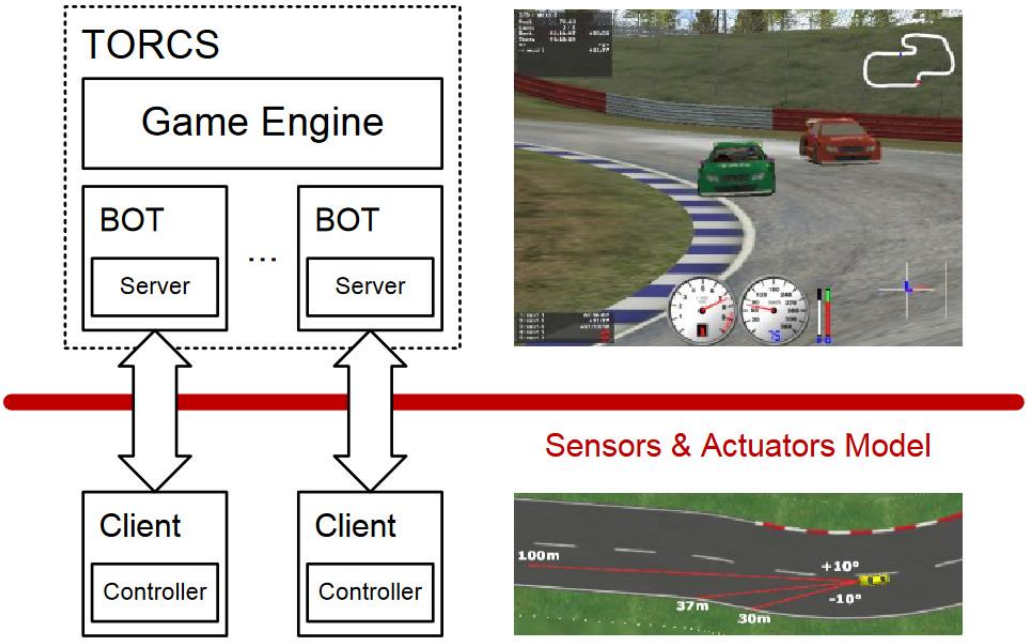
```
./configure
make
sudo make install
sudo make datainstall
```

- Kurulum tamamlandıktan sonra "`torcs`" veya "`torcs -vision`" komutu ile kurulumun doğruluğu test edilebilir.

4.3.2 Simülasyon Kurulumu ve Yapılandırılması

Bu alt başlıkta çalışma için yapılan kurulum hakkında kısa bilgi verilmiştir. TORCS simülatörü bir GNU/Linux işletim sistemi olan Ubuntu 20.04 üzerine

kurulmuştur. 20.04 versiyonunda GCC kütüphanesinden dolayı kurulumda hata ile karşılaşılmaktadır fakat simülatör kodları bu çalışma kapsamında kullanılabilir hale getirilerek, hata alınan satırlarda gerekli düzenlemeler yapılarak uyumluluk sorunları giderilmiştir. Eğitim sürecinde her bir algoritmanın modeli 4000 bölüm (episode) eğitileceği için süreç çok uzun olacaktır. Simülatör içerisinde el ile simülasyon hızı ayarlanabilmektedir ancak belirli sürelerle simülasyon sırasında RAM hatası almamak için yeniden başlatılmaktadır. Her bölümün (episode) başlangıcında bunu yapmamak için TORCS kodlarına müdahale edilerek "-j" bayrağı eklenmiş ve terminal üzerinden TORCS başlatılırken bu bayrak ile başlatılmıştır. Bu sayede simülasyon hızı 4 kata kadar arttırılabilmiş ve eğitim süreçleri önemli ölçüde hızlanmıştır. Eğitim sürecinde araç tek başına tur atacak şekilde ayarlanmış ve belirli bir port belirlenmiştir. Bu port üzerinden UDP kullanarak simülatör ile iletişim kurulmuştur. UDP vasıtasıyla farklı portlar üzerinden oyun motoruyla birden fazla istemci, aynı anda iletişim kurabilmektedir. Bu iletişim modelini gösteren mimari Şekil 4.2'de gösterilmiştir.



Şekil 4.2: Her bir istemcinin UDP kullanarak oyun motoru ile arasındaki iletişim.

4.3.3 Simülasyon Parametreleri ve Senaryoları

Bu alt başlıkta simülasyonda kullanılan; komutlar, sensör ve kontrol bilgileri, eğitim ve yarış pisti hakkında detaylı bilgi verilmiştir.

Belirtilen simülasyon ortamı, derin pekiştirmeli öğrenme modellerinin geliştirilmesi ve test edilmesi için kritik bir araçtır. Gerçek zamanlı olarak simülasyon içinde test edilen modeller, algoritmaların performansını detaylı bir şekilde analiz etme imkânı sunar. Simülasyon işlemleri modelin öğrenme sürecini ve karar verme mekanizmalarını gözlemlemek için iyi bir fırsat sunar ve XAI tekniklerinin uygulanabilirliğini artırır.

Simülasyon için oyunda yer alan arkadan itişli (RWD) bir yarış arabası tercih edilmiştir. Araç antrenman modunda (TORCS içerisinde Practice mod) eğitilmiş ve gerekli durumlarda simülasyonu sonlandırmak, yeniden başlatmak için gerekli konfigürasyonlar yapılmıştır. Test yarışı için ise hızlı yarış modu (TORCS içerisinde Quick Race) olan birden fazla aracın katılabildiği yarış türü seçilmiştir. Burada 3 farklı araç aynı anda yarıştırılmıştır. Modeller eğitim sürecinde eğitim modunda yarıştırılmış daha sonra her bir algoritmanın en iyi modelleri aralarında yarış modunda yarıştırılmıştır.

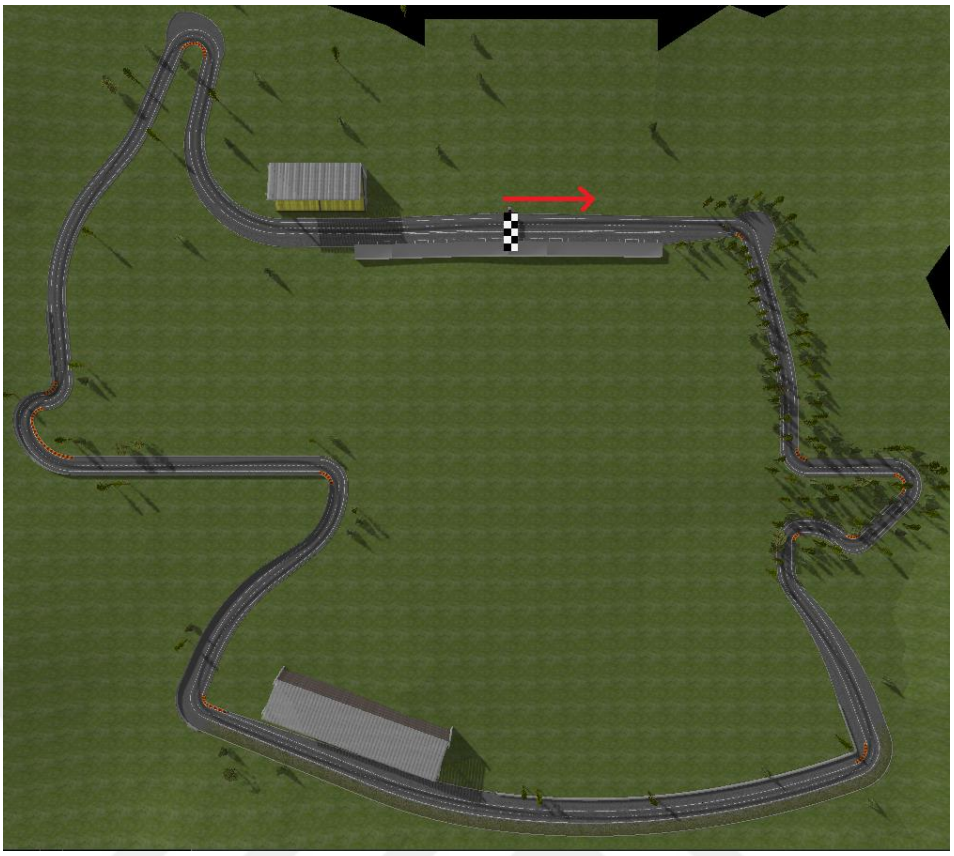
Eğitim aşamasında seçilecek pist de büyük önem arz etmektedir. Çeşitli sürüş ve pist koşullarını test etmek için simülasyon içerisinde bulunan pistler incelenmiştir. Bu doğrultuda Şekil 4.3'te gösterilen Aalborg pisti tercih edilmiştir. Aalborg pisti 14 virajdan ve 3 düzlükten oluşmaktadır. Yüksek hızlı virajları, uzun düzlükleri, sert virajları, aracın pist dışına çıktığı durumlarda lastik yüzeylerinin çim ile temas etmesi sonucu araç dinamiğinin etkilenmesi, eğimli yollar ile aracın iniş ve tırmanış esnasındaki farklı koşullardaki dinamikleri ve yakın duvarları ile model eğitimine önemli katkılar sağlayacağı düşünülmüştür. Bu özellikler, algoritmanın çeşitli sürüş koşullarına adaptasyonunu test etmek ve geliştirmek için ideal bir ortam sunmuştur. Aalborg pistinin seçilmesi çeşitli ve zorlayıcı sürüş koşulları sunarak, sürüş algoritmalarının gerçek dünya senaryolarına olan adaptasyonunu kapsamlı bir şekilde test etme fırsatı sağlamıştır. Yüksek hızlı virajlar aracın dinamik sürüş kabiliyetlerini değerlendirmekte, sert virajlar ise frenleme ve hızlanma kararlarının doğruluğunu sınamakta katkı sağlamıştır. Ayrıca pist kenarındaki çim alanlar, aracın kontrol dışı

taşma durumlarında nasıl tepki verdiğini görmek için ideal bir ortam sunarken yakın duvarlar algılama ve manevra yeteneklerini ölçmede kullanılmıştır. Pistin belirtilen bu özellikleri, modelin her yönünü ayrıntılı bir şekilde test etme ve geliştirme imkânı sunmuş aynı zamanda algoritmanın güvenlik ve etkinliğini artırmak için değerli veriler sağlamıştır.

Aalborg pisti, karmaşık bir yapıya sahip ve çeşitli sürüş becerilerini test eden bir pist olarak dikkat çekmektedir. Genel yapısı uzun düzlükler ve teknik virajlar içermesiyle hem hızlı sürüş becerilerini hem de viraj alma tekniklerini ön plana çıkarmaktadır. Pistin viraj sayısının fazla olması ve bu virajların çeşitliliği (farklı açılarda sert ve yumuşak virajlar, yüksek hızda dönülebilecek virajlar vs.), otonom sürüş algoritmaları için önemli bir test ortamı oluşturmaktadır.

Pist üzerindeki düzlükler, araçların maksimum hızlara ulaşabilmeleri için fırsat sunarken; bu bölümler, araçların hızlanma ve gaz kontrolü becerilerini değerlendirme imkânı sağlar. Özellikle başlangıç düzlüğü ve 14. virajdan 1. viraja geçiş, yüksek hız testi için idealdir.

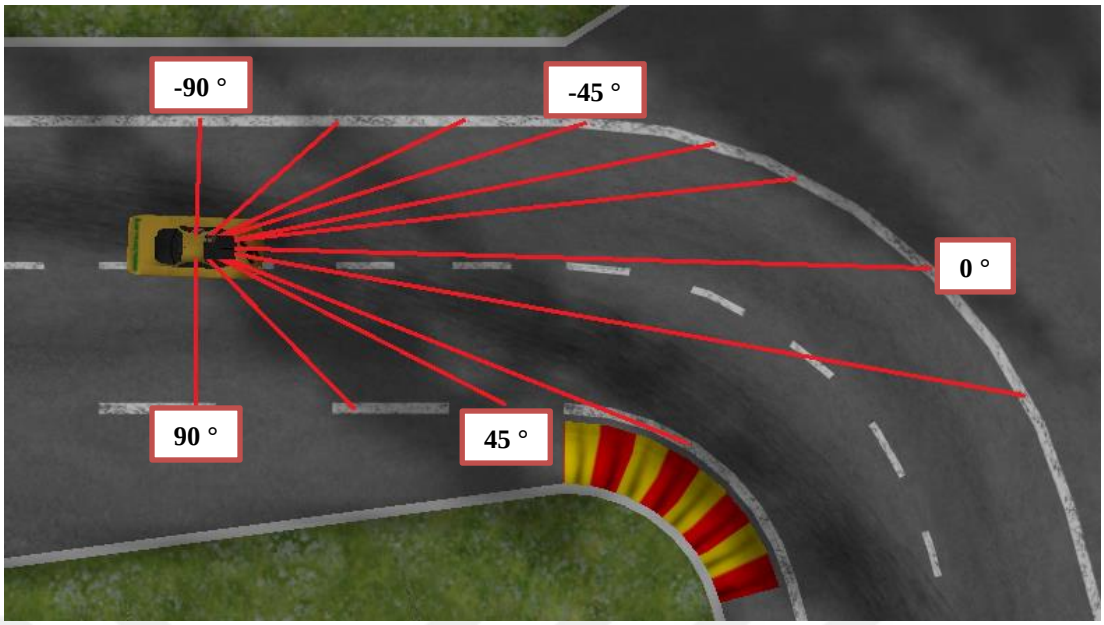
Aalborg pistindeki virajlar dar ve keskin dönüşlerden geniş yarıçaplı dönüşlere kadar çeşitlilik göstermektedir. Örneğin 1, 2 ve 3 numaralı virajlar keskin dönüşler içerirken; 11 ve 12 numaralı virajlar daha geniş açılara ve yüksek hızlı geçişlere izin verir. Bu çeşitlilikler direksiyon kontrolü, frenleme ve hız kontrolleri gibi algoritmaların farklı yönlerini test etmek için oldukça idealdir. Her virajın kendine özgü karakteri, algoritmaların adaptasyon yeteneklerini test eder. Özellikle pist dışına çıkma riski taşıyan keskin virajlar algoritmaların sınırlarını zorlar. Böylelikle virajları alma sırasında meydana gelen çarpışma ve pistten çıkma gibi olaylar algoritmanın kusurlarını ve iyileştirilmesi gereken alanları belirlemek için kullanılabilir. Bu pistteki algoritmaların performansı, gerçek dünya koşullarına ne kadar iyi uyum sağladıklarını ve çeşitli sürüş senaryolarına yanıt verdiklerini anlaşılmasına yardımcı olabilir. Aalborg pistinin bu özellikleri ile DRL modellerinin performansını kapsamlı bir şekilde değerlendirme ve gerekli optimizasyonları yapmak için önemli bir ortam sağlaması hedeflenmiştir.



Şekil 4.3: Aalborg Pistinin Kuş Bakışı Görünümü.

Sensör yapılandırması yapılırken aracın pist limitlerine olan uzaklıklarını belirli açılarla taradığı sensör bilgisi olan (bkz. Tablo 4.2) "track" isimli sensörün yaptığı tarama açıları (4.1) eşitliğindeki vektörde verilmiştir. Sensör araçtan belirlenen açılarda ışın çizerek bu ışınların uzunluklarını baz alarak değer döndürür. Bu sadece pozitif açılarda yapacağı taramayı göstermektedir. Ayrıca negatif açılarda da aynı taramayı yapmaktadır. Bu taramaya ilişkin temsili bir görsel Şekil 4.4'te verilmiştir.

$$Açı\ Vektörü = [0 \quad 0.5 \quad 1 \quad 1.7 \quad 2.5 \quad 4 \quad 7 \quad 12 \quad 19 \quad 45] \qquad (4.1)$$



Şekil 4.4: Açı vektörü girildiğinde yapılan taramanın temsili görüntüsü. Aracın merkezinden vektördeki bulunan açılarda ışınlar çizilir ve pist limitlerine olan uzaklıkları döndürür.

Kullanılan diğer sensör bilgileri (bkz. Tablo 4.2'deki bazı sensörler); aracın orta şerit ile olan açı farkını belirten "angle", aracın mevcut tur süresini belirten "curLapTime", aracın ne kadar hasar aldığını ya da hasarsız olduğunu belirten "damage", aracın başlangıç noktasından olan uzaklığını belirten "distFromStart", toplam katedilen mesafe "distRaced", aracın hızına ait bilgi "SpeedX", aracın yanal ve dikey hızlarını belirten "SpeedY" ve "SpeedZ", motor devrini temsil eden "rpm", bir önceki başarılı tur süresini temsil eden "lastLapTime", aracın şerit üzerindeki konumunu (eğer araç şeridin tam ortasındaysa 0, şerit limitini ihlal ettiğinde -1'den küçük veya 1'den büyük) ifade eden "trackPos", her bir tekerin dönüş hızı "wheelSpinVel" şeklindedir. Bu sensör bilgilerden "angle", "track", "trackPos", "SpeedX", "SpeedY", "SpeedZ", "wheelSpinVel" ve "rpm" durum uzayı olarak belirlenmiş ve yapay sinir ağlarına girdi olarak verilmiştir. Girdi olarak verilirken bu değerler normalize edilmiştir. Aracı kontrol etmek için modele üç farklı girdi uygulanmıştır. Bunlar direksiyon açısı, gaz ve frendir. Direksiyon açısı [-1,1] aralığında, gaz ve fren aralığı [0,1] aralığında olup bu eylemler sürekli eylem uzayındadır. Direksiyon eylemi -1 olduğunda araç direksiyonu 1 tur sola çevirmiş olacaktır. Bu değer sıfır iken direksiyon ve teker açıları sıfır derecedir ve araç düz hareket edecektir. Değerin +1 olması durumunda ise direksiyon tam tur sağa

döndürülmüştür. Bu da aracın ilerlemesi durumunda sağa doğru döneceği anlamına gelir.

Eğitim esnasında `-nolaptime` komutu kullanılmıştır. Bu sayede eğitim sürecinin başlarında aracın çok yavaş hareket etmesine bağlı keşfinin önüne geçmekten kaçınılmıştır. Eğitim sürecinde ilk başta `-nofuel` komutu kullanılmıştır. Ancak bu komuta bağlı olarak aracın gerçek yarış senaryolarında, ağırlığının zaman içerisinde yakıt tüketimine bağlı değişmesine bağlı olarak fiziksel modeli de değişeceği için modelin düzgün karar vermesini etkilediği gözlemlenmiştir. Bu nedenle eğitim süreci bu komut kullanılmadan tekrarlanmıştır. Ek olarak `-j 0.25` komutu kullanılarak simülasyonun 4 kat hızlandırılması sağlanarak, model eğitim süreçleri hızlandırılmış; normal şartlarda her bir model için sürecek süre yaklaşık 5 gün iken 30 saat civarına kadar indirgenmiştir.

4.4 DRL Algoritmalarının Model Eğitimi ve Test Stratejileri

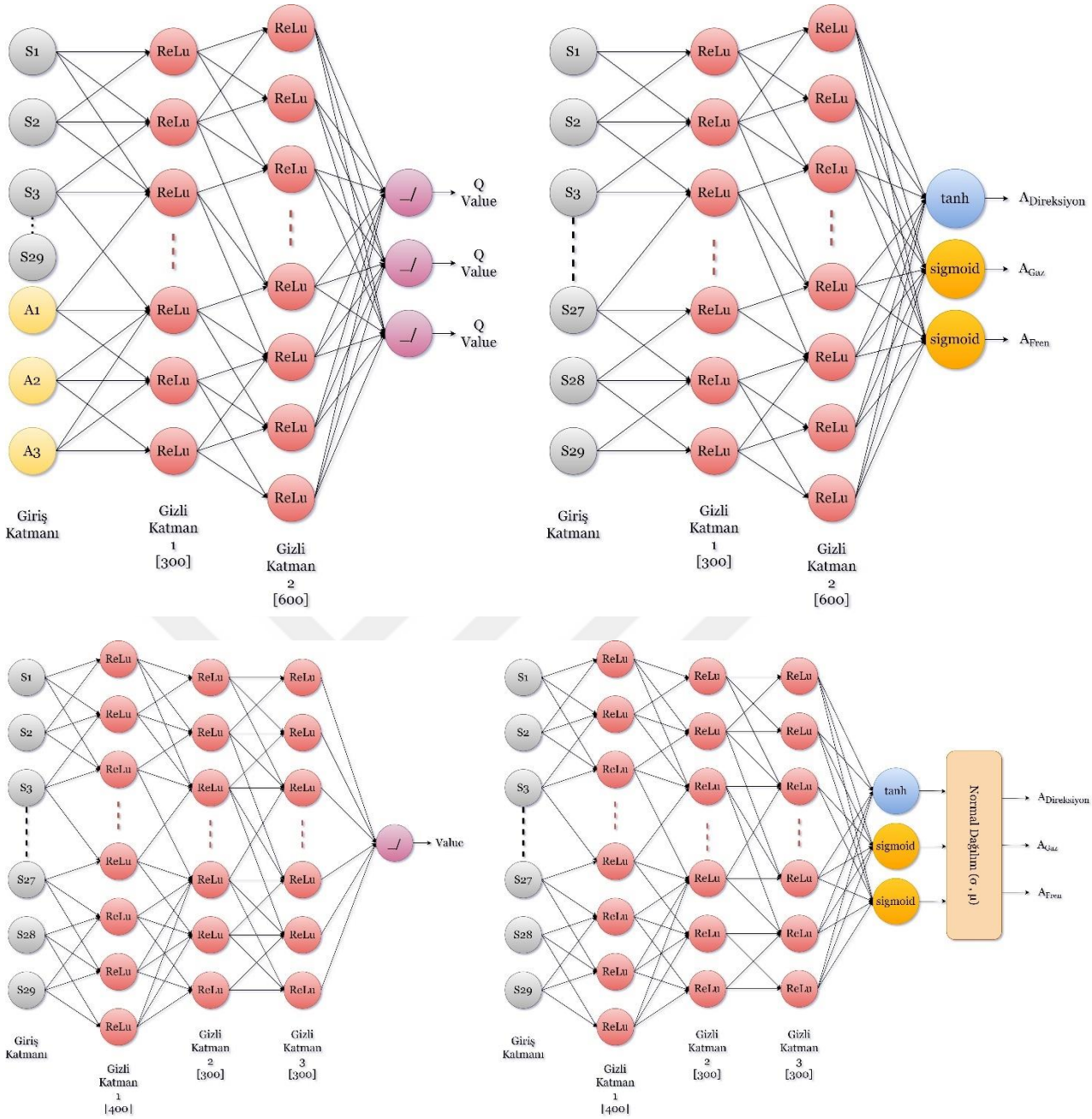
Bu alt bölümde, XAI ve DRL algoritmaları kullanılarak otonom araç sürüşü için geliştirilen modellerin eğitimi ve uygulama süreçleri ele alınmıştır. TD3, DDPG ve PPO olarak üç farklı derin pekiştirmeli öğrenme modeli kullanılmıştır. Bu modeller TORCS simülasyon ortamı kullanılarak; Aalborg pisti üzerinde, 4000 bölüm (Episode) boyunca eğitilmiştir. Büyük hata yapıldığında ya da başarıyla turu tamamladığında bölüm sonlandırılmış ve bir sonraki bölüme geçilmiştir. Eğitim esnasında belirli aralıklarla modeller kaydedilmiş ve ajanların elde ettiği istatistikler kaydedilmiştir. Ardından eğitilen TD3, DDPG ve PPO modelleri 25 turluk bir test yarışına tâbi tutulmuştur. Eğitim esnasında ve test yarışında elde edilen sayısal veriler Bölüm 5'te detaylı şekilde ele alınarak değerlendirilmiştir; ayrıca XAI yöntemleri de bu bölümde işlenmiştir.

4.4.1 Model Mimarileri

Çalışma kapsamında 3 farklı DRL algoritması modeli eğitilmiştir. Otonom sürüş için; simülatörde ve gerçek hayatta, durum uzayı (State space) ve eylem uzayı (Action space) sürekli zamanlıdır. Önceki çalışmalar tercih edilen algoritmalar

değerlendirilmiş ve kullanılacak olan DRL algoritmaları DDPG, TD3 ve PPO olarak belirlenmiştir. Bu modeller otonom sürüş için uygulanabilir hale getirilmiştir. Üç modele ait hiperparametreler Tablo 4.4'te verilmiştir. DDPG ve TD3 modellerine ait kritik ve aktör ağlarının (Critic & Actor Networks) mimari Şekil 4.5-a'da verilmiştir. İlk gizli katman 300 düğüm ikinci gizli katman ise 600 düğümden oluşmaktadır. Aktivasyon fonksiyonlarında ise ReLU tercih edilmiştir. Aktör ağının çıktısında direksiyon açısı eylemi için tanh aktivasyon fonksiyonu, gaz ve fren için ise sigmoid fonksiyonu kullanılmıştır.

PPO ağının mimarisi ise Şekil 4.5-b'de verilmiştir. Diğer ağ modellerine göre çıktı katmanında mu (ortalama) ve sigma (standart sapma) kullanılmıştır. Sürekli eylem uzayı genellikle olasılık dağılımı ile ifade edilir. Aktör ağının mu (μ) çıkış katmanında direksiyonu -1 ile 1 arasında sınırlamak için tanh gibi uygun aktivasyon fonksiyonları kullanılarak; gaz ve fren için çıkışı 0 ile 1 arasında sınırlamak için sigmoid fonksiyonu kullanılarak hesaplama yapılmıştır. Ardından sigma (σ) çıkış katmanı (standart sapma) politikanın dağılımı ayarlanmıştır.



Şekil 4.5: a) DDPG ve TD3 algoritmalarının kritik ve aktör ağlarının mimarisi **b)** PPO algoritmasının ağ mimarisi.

Tablo 4.4: Modellere ait hiperparametreler.

Algoritma	Learning Rate Actor & Critic	Batch Size	Discount Factor	Clip Rate
TD3	0.001	64	0.99	-
DDPG	0.001	64	0.99	-
PPO	0.0003	64	0.95	0.20

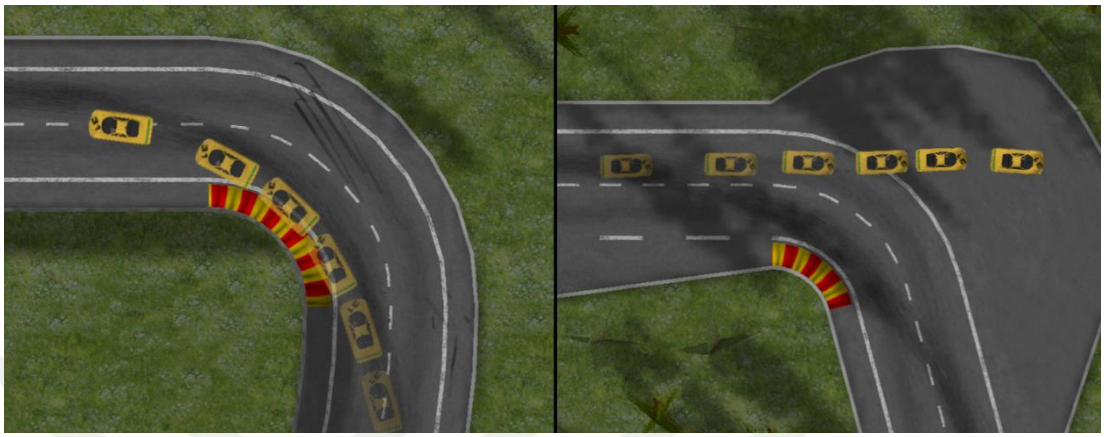
4.4.2 Ödül Fonksiyonu ve Eğitim Stratejisi

Derin pekiştirmeli öğrenme modellerinin başarılı bir şekilde eğitilmesinde ödül fonksiyonunun belirlenmesi ve etkin bir eğitim stratejisinin uygulanması büyük önem arz ettiği önceki bölümlerde görülmüştür. Bu nedenle otonom sürüş için performans ve güvenlik gibi kriterler göz önünde bulundurularak bir ödül fonksiyonu belirlemek gerekir. Bu fonksiyon aracın çevresel etkileşimleri ve alınan kararların sonuçlarına göre pozitif ya da negatif geri bildirimlerle modeli cezalandırır veya ödüllendirir. Ödül fonksiyonunun doğru bir şekilde tasarlanması, modelin istenilen davranışları öğrenmesini ve gereksiz ya da zararlı eylemlerden kaçınmasını sağlar. Eğitim stratejisi ise modelin verimli bir şekilde eğitilmesi için kullanılan yöntemleri içerir. Bu bölümde kullanılan ödül fonksiyonunun detaylarına ve otonom sürüş için modelin nasıl etkin bir şekilde eğitildiğine dair stratejiler ele alınmıştır.

Tablo 5.3'te yer alan sonlandırma kriterleri 4000 bölümden oluşan eğitim için her bir bölümün sonlandırıp bir sonrakine geçmesini ifade eder. Eğer araç eşik değerden yüksek bir hasar alırsa kaza (major collision), pist limitleri dışına taşar ve bir süre sonra piste geri dönmezse (out of track), araç belirli bir süre çok yavaş hareket eder ya da tamamen hareketsiz kalırsa (no progress), araç spin atar veya ters yönde ilerlemeye başlar ise (spin or wrong direction), turu başarılı bir şekilde bitirirse (successful); bölüm sonlandırılır ve negatif ödül/ceza alır. Ardından bir sonraki bölüm başlatılır. Başarılı olduğu durum hariç büyük hata (Major Fail) yapmış varsayılır. Eğer kazadan dolayı alınan hasar eşiğin altındaysa, araç anlık olarak pist limiti ihlali yapmış ve tekrar piste dönmüşse, kısa süreliğine hareketsiz kalmış ve sonra ilerlemeye devam etmiş ise ya da spin atarken hızlı bir şekilde toparlamış ise bu durumlar küçük hata (Minor Fail) olarak geçer. Bu koşullarda geçerli bölüm sonlandırılmaz ve model sadece negatif ödül alır. Modeller eğitilirken bu sonlandırma kriterleri sayesinde çeşitli durumlarda erken sonlandırma yaparak, eğitim sürecinde zaman tasarrufunun sağlanması ve verimin arttırılması hedeflenmiştir.

Aracın pist limitlerini ihlal ettiği iki farklı durum Şekil 4.6'da verilmiştir. Şekil 4.6-a'da sonlandırma kriterlerine uymadığı için bölüm sonlandırılmamıştır. Ancak Şekil 4.6-b'de araç pist limitlerini aştıktan sonra yeterli sürede piste geri dönmemiş ve bu nedenle büyük hata kabul edilerek bölüm sonlandırılmıştır ve negatif ödül almıştır.

Bir diğer sonlandırma kriterine ait görsel Şekil 4.7'de verilmiştir. Burada araç hızla girdiği virajda hakimiyetini kaybederek spin atmış ve bölüm sonlandırılarak negatif ödül almıştır.



Şekil 4.6: a) Aracın pist limitlerini kısa süreli aşması **b)** Aracın pist limitlerini uzun süreli ihlal etmesi.

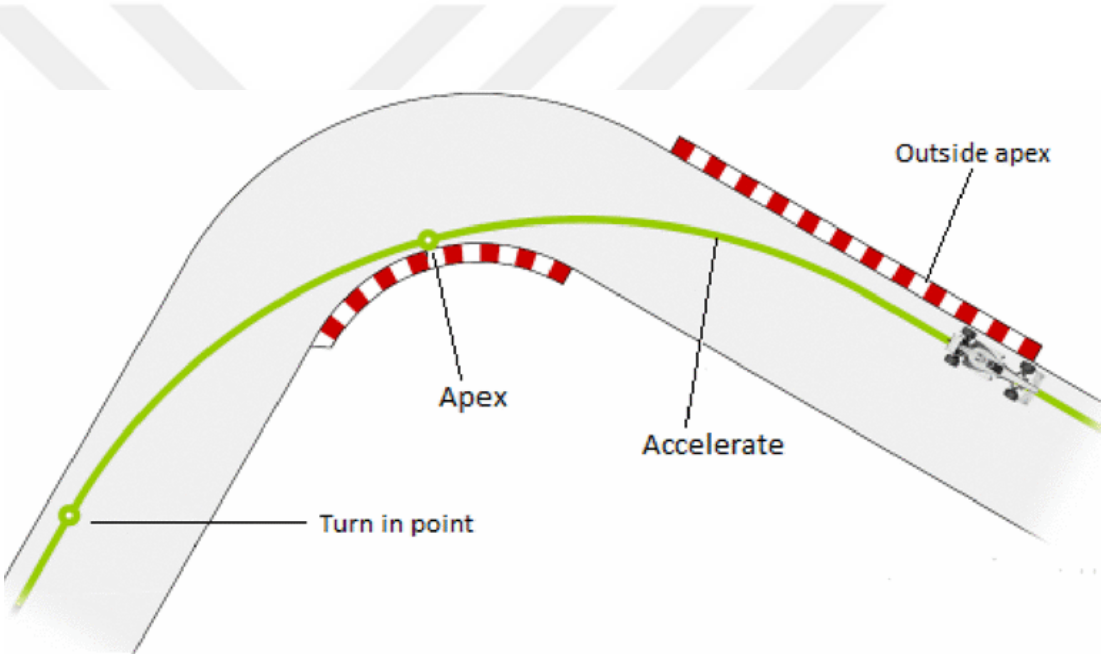
Şekil 4.7: Aracın kontrolü kaybederek spin attığı senaryo.

Ödül fonksiyonu (4.2) eşitliğindeki gibi üç parçadan oluşmaktadır. R_p ceza olup negatif değerlidir ve R_c pozitif olup turun başarılı tamamlanmasını ifade eder. R_t ise her adımda modele verilen negatif ceza puanıdır. R_p kaza, pist dışına taşma, spin atma, ters yönde ilerleme, yavaş ilerleme veya çok fazla yavaşlama durumlarında verilen cezadır. Her bir negatif olay için farklı ceza puanı ayarlanmıştır. İlk terim olan R_a aracın hızı ve şerit ile arasındaki uzaklıktan türetilmiştir. Bu ödül belirlenirken gerçek hayattaki yarışlar örnek alınmıştır. Yarış araçlarının, yarış esnasında virajları en hızlı bir şekilde tamamlayabilmesi ve aracın viraj çıkışlarında maksimum hıza ulaşabilmesi hedeflendiği için pist şeridin ortasından gitmesi beklenmez. Araç viraja girerken, virajı mümkün olduğunca dışarıdan (turn in point kısmı) almalı ve çıkış hızını (outside apex) olabildiğince yüksek tutması gerekir. Aracın belirtilen bu hareketi ve izlediği yol ideal yarış çizgisi (ideal racing line, apex) olarak tanımlanmaktadır (Smith 1978). Örnek senaryo Şekil 4.8'de gösterilmiştir. Fakat aracın her durumda pist

limitlerine yakın performans göstermesi ya da pist limitlerini aşacak bir durum oluşturması istenmemektedir. Belirtilen nedenlerden dolayı (4.3) eşitliğinde belirtilen fonksiyon tasarlanmıştır. Burada aracın mümkün olduğunca hızlı hareket etmesi ve aynı zamanda mümkün olduğunca pist sınırları içerisinde kalarak aracın yüksek performansını güvenli bir şekilde sağlaması hedeflenmiştir.

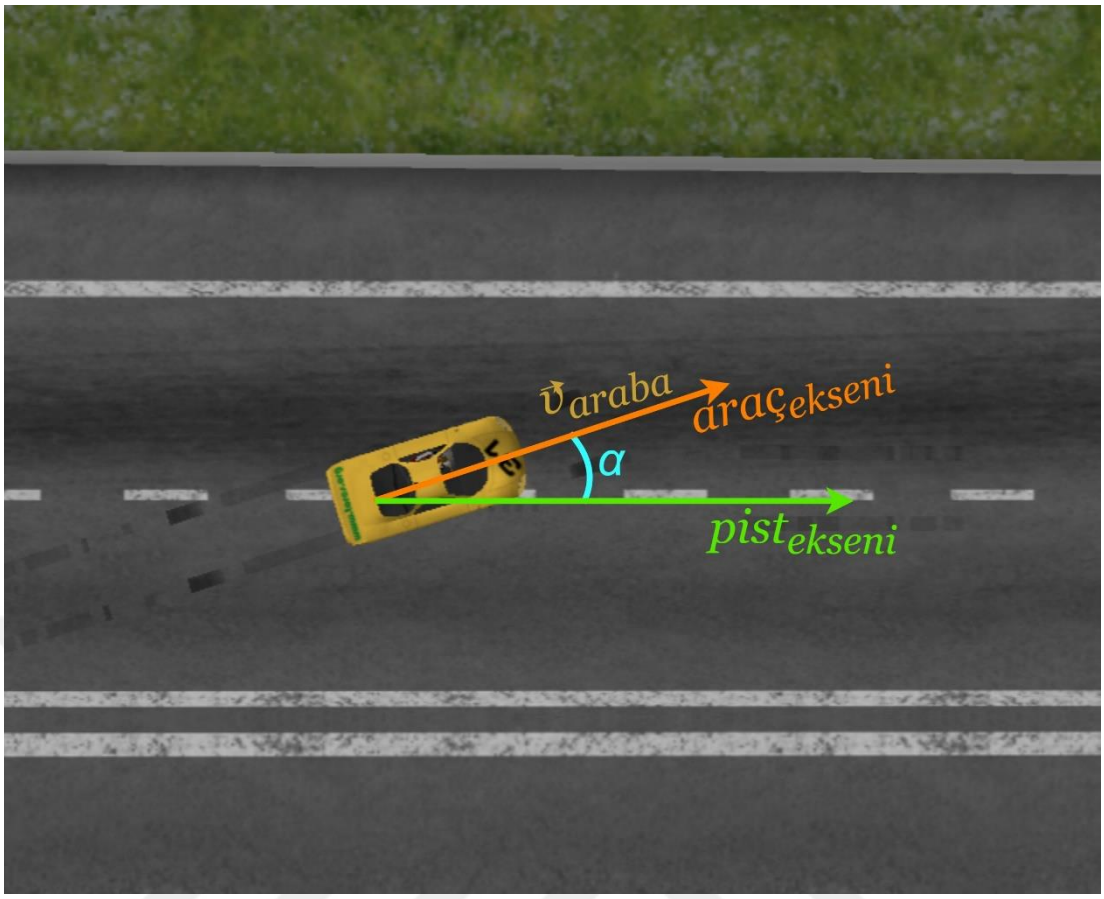
$$R = R_a + R_p + R_c + R_t \tag{4.2}$$

$$R_a = V \cdot \cos(\alpha) - |V \cdot \sin(\alpha)| - V \cdot |t_{pos}| \tag{4.3}$$



Şekil 4.8: İdeal yarış çizgisi örneği(Bugeja ve diğ. 2017).

Eşitlik (4.3)'teki V araç hızını, α araç ekseni ile pist ekseni arasındaki açıyı, t_{pos} değeri ise aracın pist üzerindeki yanal konumunu ifade eder. Bu duruma ilişkin temsili görsel Şekil 4.9'da yer almaktadır. Eğer araç iki şeridin tam ortasındaysa sıfıra eşittir. Sağ şeridin tam üstü 1 değerine eşitken sol şeridin tam üstünde olması durumunda -1 değerine eşittir. Mutlak değer olarak 1'den büyük olduğunda aracın pistin dışına taştığını ifade etmektedir. t_{pos} TORCS'un döndürdüğü bir referans değeridir ve aracın pist üzerindeki konumunu ifade etmektedir.



Şekil 4.9: Araç ile pist ekseni arasındaki açı.

4.4.3 Teknik Detaylar, Yazılımlar ve Platform

Tez kapsamındaki simülasyon ve test işlemleri TORCS motoru ile Ubuntu 20.04 işletim sisteminde çalıştırılmıştır. Model eğitimi ve test süreçleri için Python3.7 programlama dili tercih edilmiş ve bu süreçlerde TensorFlow, Keras, NumPy, gym-torcs ve PyTorch gibi çeşitli kütüphanelerden faydalanılmıştır. Eğitim için NVIDIA GeForce GTX 1080Ti GPU, Intel i7-6850K CPU ve 16 GB RAM donanıma sahip bir bilgisayar kullanılmıştır. Her bir modelin 4000 bölümde eğitimi yaklaşık 32 saat sürmüştür.

5. DENEYSEL UYGULAMA SONUÇLARI VE BULGULAR

Bu bölümde, tez kapsamında kullanılan DRL modellerinin (TD3, DDPG ve PPO) performans değerlendirmeleri ve elde edilen bulgular detaylı bir şekilde incelenmektedir. Modeller Aalborg pisti üzerinde gerçekleştirilen 4000 eğitim bölümü ve 25 turluk test yarışı sürecinde değerlendirilmiştir. Eğitim ve test süreçlerinde modellerin karşılaştığı zorluklar, başarılar ve öğrenme eğilimleri, çeşitli metriklerle analiz edilmiştir. Bunlardan bazıları kazandıkları ödüller, ortalama hız, ortalama ödül, başarı oranı, ihlal ve büyük hata sayıları, maksimum ödül gibi metriklerdir. Bu analizler ile modellerin sürüş stratejilerini ve karşılaştıkları senaryolara adaptasyon kabiliyetlerinin anlaşılmasını sağlamak amaçlanmıştır. Ayrıca eğitim sürecinde kaydedilen verilere dayanarak modellerin performansını ölçen çeşitli istatistiksel veriler ve grafikler yer almaktadır. Ödül fonksiyonlarının, modellerin davranışları üzerindeki etkisi, yapılan test yarışlarında modellerin gösterdiği performansla birlikte değerlendirilerek; her bir modelin güçlü ve zayıf yönleri ortaya konulmuştur. Bu bölümde sunulan analitik bulgular otonom araçların gerçek dünya senaryolarına adaptasyonu açısından kritik öneme sahip olması hedeflenmiştir.

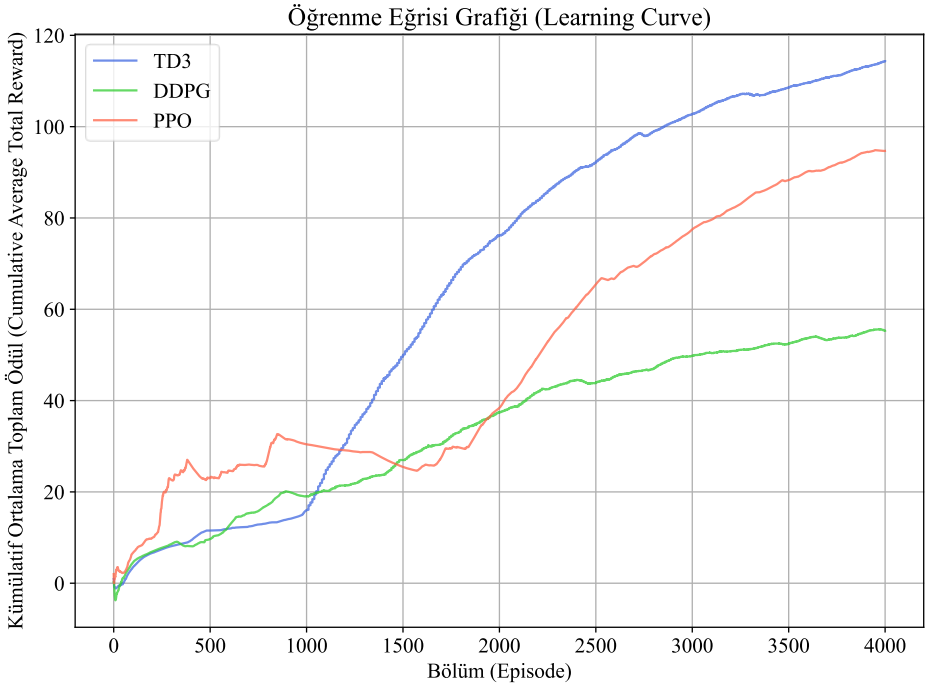
Ayrıca DRL modellerinin karmaşık ve dinamik çevrelerde nasıl kararlar aldıklarını açıklamaya yönelik yöntemler üzerine de odaklanılmıştır. Eğitim esnasında gerçekleşen kaza, ihlal gibi olayların ısı haritaları; test yarışından elde edilen verilerden üretilen SHAP değerleri ile analiz edilerek, XAI açısından değerlendirilmiştir. SHAP değerleri ile global ve lokal açıklanabilirlik üzerinde durulmuştur.

5.1 DRL Algoritmalarının Eğitimine Ait Sayısal Sonuçlar

Bu alt bölümde modellerin eğitim sürecinde elde ettikleri istatistikler, tablo ve grafik şeklinde sunulmuş; bu sayısal veriler üzerinden sonuçlar analiz edilmiştir. Kullanılan DRL modelleri olan TD3, DDPG ve PPO'nun performans değerlendirmeleri kapsamlı bir şekilde irdelenmiş, modellerin eğitim sürecindeki öğrenme dinamikleri dikkate alınarak başarıları ve zorlukları detaylıca incelenmiştir. 4000 eğitim bölümü boyunca toplanan veriler, modellerin karşılaştıkları senaryolara

göre adaptasyon kabiliyetlerini ve sürüş stratejilerinin anlaşılmasını sağlamıştır. Eğitim sürecinde elde edilen istatistikler ve çeşitli metrikler, her bir modelin güçlü ve zayıf yönlerini ortaya koymakla birlikte bu modellerin gerçek dünya senaryolarında adaptasyon süreçleri bakımından kritik öneme sahiptir.

4000 bölümden oluşan eğitim sürecinin sonunda 3 algoritmaya ait ödüllerin kümülatif ortalama ödülleri hesaplanmıştır. Eğitim sürecinin öğrenme eğrisi grafiği Şekil 5.1'deki gibidir. TD3 algoritması başlangıçta en hızlı öğrenme ivmesine sahip olmasa da zamanla sürekli olarak performansını artırmıştır. 4000 bölüm sonunda en yüksek kümülatif ortalama toplam ödülüne ulaşmıştır. Bu TD3'ün kararlılık ve performansının diğer iki modele göre daha üstün olduğunu gösterir. DDPG algoritması, TD3'e benzer bir başlangıç yapmış ancak zamanla performansı TD3'ün altında kalmıştır ve TD3 kadar yüksek bir toplam ödül değerine ulaşamamıştır. DDPG'nin öğrenme eğrisinin daha düşük olması algoritmanın belirli durumlarda daha dikkatli karar vermesi ya da daha yavaş öğrenmesiyle ilgili olabilir.



Şekil 5.1: Algoritmaların öğrenme eğrisini gösteren kümülatif ortalama ödüllerin bölümlere göre dağılımının grafiği.

PPO başlangıçta çok iyi bir performans göstermiş ancak 800. bölümden sonra bir müddet öğrenme ivmesi negatif olmuştur. Yaklaşık 1500. bölümden itibaren iyi bir ivme yakalayarak ikinci sıraya yükselmiştir. TD3 modelinin gerisinde kalsa da

DDPG'ye göre daha iyi bir performans sergilediğinden söz edilebilir. Bu grafiğe göre TD3'ün DDPG'ye göre daha çok ödül toplaması 2 farklı kritik ağı kullanarak gürültüye karşı daha az duyarlı olmasından kaynaklandığı düşünülmüştür. PPO'nun erken dönemde hızlı öğrenmesi keşifte başarılı olduğunu göstermekte ardından düşüşe geçmesi ise lokal minimuma sıkışmasından kaynaklanabileceği düşünülmüştür.

Tablo 5.1 TD3, DDPG ve PPO algoritmalarının eğitim sırasında elde edilen önemli istatistikleri özetlemektedir. Tablodaki ortalama bölüm süresi, kazanılan ödül ve hız değerleri %95 güven aralığı ile verilmiştir. Ortalama yarışılan mesafe metriği modellerin ortalama olarak ne kadar mesafe kat ettiğini göstermektedir. TD3 ve PPO, DDPG'ye göre daha uzun mesafeler kat etmiştir; bu da TD3 ve PPO'nun, pistte daha uzun süre büyük hata yapmadan kaldığını göstermektedir. Ortalama bölüm süresine bakıldığında PPO'nun diğer iki modele göre çok daha yüksek bir ortalama bölüm süresine sahip olması, modelin bölümleri daha uzun süre sürdürebildiğini veya simülasyon içinde daha yavaş ilerlediğini işaret etmektedir. Ortalama kazanılan ödül modellerin bir bölümde ortalama olarak kazandığı ödül miktarını göstermektedir. TD3 en yüksek ortalama ödülle performansını diğerlerinden daha üstün bir şekilde göstermiştir. DDPG ise bu alanda oldukça düşük bir performans sergilemiştir.

Tablo 5.1: Model eğitimi sürecinde elde edilen temel performans istatistikleri.

	TD3	**DDPG**	**PPO**
Ortalama Yarışılan Mesafe (m)	1635	1154	1595
Ortalama Bölüm Süresi (s)	61.58±1.10	51.54±1.11	119.18±2.6
Ortalama Kazanılan Ödül	114.34±2.39	55.32±1.5	94.69±1.33
Ortalama Hız (km/h)	87.37±0.50	73.01±0.65	47.99±0.32
Başarılı Tur Oranı (%)	49.5	21.2	52.1
Toplam Başarılı Tur Sayısı	1980	849	2085
Büyük Kaza Oranı (%)	32	21	36

Ortalama hız modellerin eğitim esnasındaki ortalama araç hızını ifade etmektedir. TD3 yine bu alanda da lider konumda iken PPO'nun oldukça düşük bir ortalama hıza sahip olduğu gözlemlenmiştir. Bu değişken tek başına PPO'nun daha temkinli veya optimum hız profilini sürdürme stratejisi izlediğini ifade edebilmektedir. Başarılı tur oranı 4000 bölümde başarıyla bir turu tamamlanma yüzdesini göstermektedir. PPO bu alanda %52.1 ile en yüksek başarı oranına sahipken, DDPG

oldukça düşük bir oranla sınırlı kalmıştır. TD3 ise PPO'ya yakın bir oran elde etmiştir. Büyük kaza oranı her bir modelin karşılaştığı büyük kazaların oranını ifade etmektedir. Büyük kaza ifadesi aracın belirli bir hasar puanının üstünde hasar almasını belirtmektedir. Örneğin aracın duvara küçük sürtmesi küçük bir kaza sayılırken yüksek hızda doğrudan duvara çarpması büyük kaza olarak nitelendirilmiştir. PPO en yüksek kaza oranına sahip olmasına rağmen yüksek başarılı tur oranı ve kazanılan ödülle riskli stratejilerinin belirli bir dereceye kadar işe yaradığını göstermiş olabileceğini düşündürmektedir.

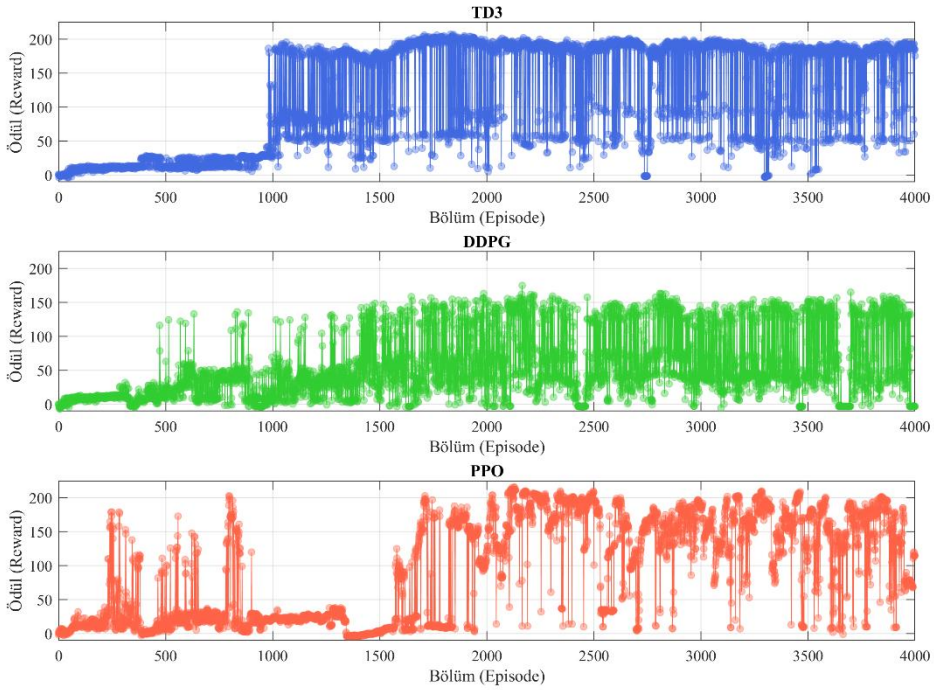
Tablo 5.2 üç modelin ilk başarılı turlarını tamamlama performansını karşılaştırmalı olarak sunmaktadır. TD3 modeli ilk başarılı turu 980. bölümde tamamlarken, DDPG ve PPO daha erken bölümlerde bu başarıya ulaşmıştır; DDPG 514. Bölümde, PPO ise en erken olan 241. bölümde başarılı bir tur tamamlamıştır. Ancak tur süresi açısından değerlendirildiğinde TD3 119.33 saniye ile en hızlı tur süresine sahipken, DDPG ile PPO sırasıyla 131.59 ve 195.21 saniye ile daha yavaş tur süreleri kaydetmiştir. Ödül miktarlarına bakıldığında ise TD3 186.55 ile en yüksek ödülü almış, PPO 164.69 ile ikinci sırada yer alırken, DDPG 124.41 ile en düşük ödülü almıştır. Bu istatistikler TD3'ün ilk başarılı turda daha etkin ve verimli performans sergilediğini, diğer modellerin ise bu aşamada daha düşük performans gösterdiğini ortaya koymaktadır.

Tablo 5.2: İlk başarılı bölüme ait istatistikler.

İlk tur tamamlandığındaki	**TD3**	**DDPG**	**PPO**
Bölüm	980	514	241
Tur Süresi	119.33	131.59	195.21
Ödül	186.55	124.41	164.69

Şekil 5.2'de yer alan grafikte üç farklı DRL modeli TD3, DDPG ve PPO için eğitim boyunca elde edilen ödül değerlerinin bölümler arasında nasıl değiştiğini göstermektedir. TD3 genelde daha yüksek ve stabil ödül değerleri elde etmiş olup, öğrenme sürecinin sonlarına doğru oldukça düzenli bir ödül yapısı sergilemiştir. PPO modeli TD3'e kıyasla daha düşük ancak yine de yüksek dalgalanmalar içermeyen bir ödül yapısı gösterirken, DDPG modeli eğitim süreci boyunca önemli dalgalanmalar

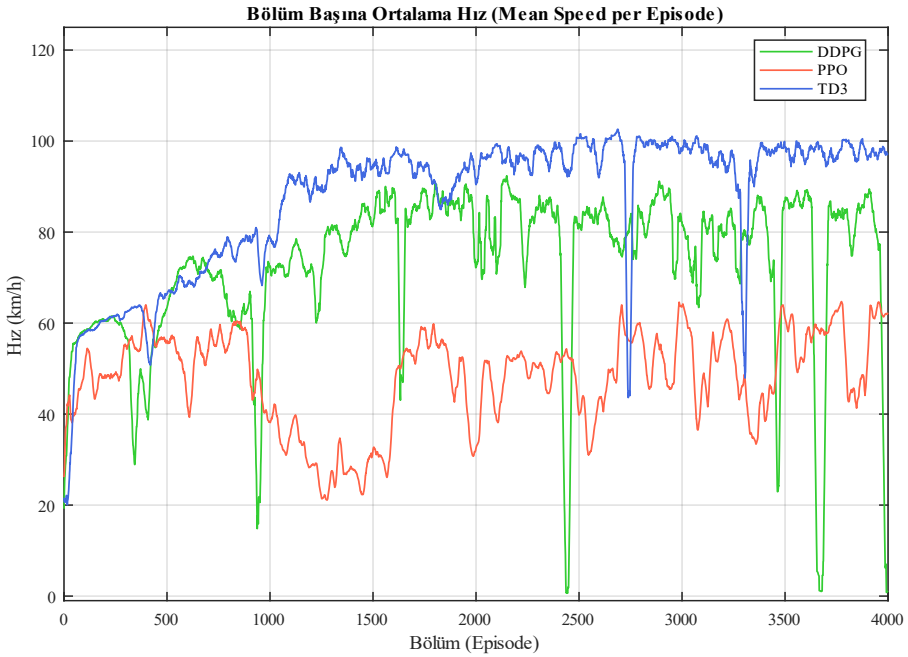
içeren ve daha düşük ödül seviyelerine işaret eden bir yapı sergilemiştir. Bu gözlemler TD3'ün diğer iki modele göre daha istikrarlı ve yüksek performans sergilediğini tekrar doğrularken, PPO ve DDPG grafiklerinde ise ödül dalgalanmalarında daha büyük zorluklar yaşadıklarını ve düşük ödül değerleri elde ettiklerine dair bulgular sunmuştur. Burada PPO ilk başarılı turlarının ardından 1000. bölümden itibaren düşüş yaşamış ve bu aşamada bir süre gelişme elde edememiştir. Bu durum lokal minimuma sıkışmasıyla açıklanmıştır. TD3 diğer modellere göre daha geç öğrenmiş ancak başarılı güncellemeleri ile ilerleyen bölümlerde istikrarlı bir sürüş ortaya koymuştur.



Şekil 5.2: Her bir bölümde elde edilen ödül grafiği.

Araçlar eğitim esnasında turlarken her bir bölümdeki pist üzerinde yol alınan toplam mesafe bölüm uzunluğunun süresine bölünerek ortalama hız değerleri elde edilmiştir. Bu ortalama hızların grafiği Şekil 5.3'te verilmiştir. TD3 genel olarak en yüksek ve en stabil ortalama hız değerlerine sahiptir, bu da modelin sürekli ve hızlı bir performans sergilediğini göstermiştir. Yeşil renkli DDPG eğrisi de yüksek hız değerleri gösterse de TD3 kadar stabil değildir ve belirli aralıklarla düşüşler yaşamıştır. PPO modeli ise özellikle başlangıç bölümlerinde düşük hız değerleriyle başlamış ve zamanla artan bir ivme göstermiş ancak diğer iki modele kıyasla daha düşük hız seviyelerinde kalmıştır. Bu hız profilleri; algoritmaların sürüş stratejilerinin, karar

verme mekanizmalarının farklılıklarını ve çeşitli sürüş koşullarına olan adaptasyonlarını yansıtmaktadır.



Şekil 5.3: Modellerin eğitim sürecindeki ortalama hızları.

Tablo 5.3 üç modelin eğitim sırasında 4000 bölümde karşılaştıkları büyük hata türlerine göre bölüm sonlandırma sayılarını göstermektedir. Bu hatalar aracın büyük hasar aldığı kaza, tekrarlayan pist limitleri ihlali, ilerlemenin olmadığı/aracın çok yavaş hareket ettiği ve spin attığı/ters yönde ilerlediği durumları içerir. Kaza (aracın hasar aldığı durumlar) tüm modeller için en yaygın sonlandırma kriteridir ve PPO modeli eğitim esnasında bu tür kazalarda en az etkilenmiş ve en az kaza yapan model olarak belirginleşmiştir. DDPG'nin bu kategoride en yüksek kaza sayısına sahip olduğu gözlemlenmiştir. TD3 ve DDPG için bu durum modellerin risk alma ve çarpışma olasılığı yüksek stratejiler geliştirdiğini gösterebilir.

Tablo 5.3: Büyük hata sonucunda bölümün sonlandırıldığı durum sayıları.

	TD3	**DDPG**	**PPO**	**Toplam**
Kaza (aracın hasar aldığı)	1267	1440	1005	3712
Tekrarlayan Pist Limiti İhlali	175	665	483	1323
İlerleme Yok / Çok Yavaş	23	304	374	701
Spin veya Ters Yön	555	742	53	1350

Tekrarlayan pist limiti ihlalleri bir aracın yarış pistinin sınırlarını aştığı, belirlenen güvenli alanın dışına çıktığı ve yeterli sürede geri dönmediği anları ifade eder. Bu kategoride DDPG diğer iki modele göre çok daha yüksek bir ihlal sayısı ile öne çıkmıştır; bu da algoritmanın aracı pist sınırları içerisinde tutma konusunda zorlandığını göstermektedir. İlerleme yok / çok yavaş kategorisi aracın belirli bir süre boyunca önemli bir ilerleme gösterememesi veya eşik hız limitinin altında hareket etmesi durumlarını kapsar. Bu kategoride PPO en yüksek sayıya sahipken, TD3 bu durumu en iyi yöneten model olarak gözlemlenmiştir. Spin veya ters yön, aracın kontrolünü kaybederek döndüğü veya yanlış yöne doğru ilerlediği durumları belirtir. PPO bu konuda oldukça düşük bir sayı ile en iyi performansı sergilemiştir. Bu hata türleri ve sonlandırma sayıları, modellerin karşılaştıkları zorlukları ve eğitim sırasında ne tür iyileştirmelere ihtiyaç duyduklarını ortaya koymakta yardımcı olmuştur. Ayrıca bu istatistikler, algoritmaların farklı sürüş senaryolarında adaptasyon yeteneklerinin değerlendirilmesinde önemli bir gösterge olarak kullanılabilir.

Eğitim sürecini değerlendirirken sadece tüm bölümler üzerinden değerlendirme yapmak doğru olmayabilir. Çünkü bir bölüm 25 adımda sonlandırılabilirken (aracın başlangıçta kaza yapması ya da ilerleme kaydetmemesi gibi durumlar) 2000 adıma kadar uzadığı durumlar da mevcuttur. Erken sonlandırmanın sebebi bölümün başlarında gerçekleşen büyük hatalardan kaynaklıdır. Bu durumda da geçerli bölüm için yeterli veri toplanmadığından analizlerde eksikliklere sebep olabilir. Bu nedenle eğitim istatistikleri değerlendirilirken modellerin başarıyla bir tam turu tamamlayabildiği durumlar ayrıca ele alınmıştır. Tablo 5.4 başarıyla tamamlanan bölümler boyunca; TD3, DDPG ve PPO modellerinin tur bazındaki performanslarını göstermektedir. Burada ortalama ödül, hız ve süre %95 güven aralığı ile verilmiştir.

Ortalama ödül tüm başarılı turlarda elde edilen ödüllerin ortalamasını ifade etmektedir. TD3 modeli 187.90±0.33 puan ile en yüksek ortalama ödülü kazanmıştır; bu da TD3'ün başarılı turları daha verimli tamamladığını göstermiştir. DDPG ve PPO modelleri ise sırasıyla 135.42±0.86 ve 160.06±1.45 puan ile daha düşük ortalama ödüller almışlardır. Maksimum ödül başarılı turlarda bir turda elde edilen en yüksek ödüldür. Bu kategoriye bakıldığında PPO'nun 216 puanla en yüksek ödülü elde ettiği görülmüştür. Bu PPO'nun belirli turlarda diğer modellere kıyasla daha üstün

performans sergilediğini gösterebilir. TD3 ve DDPG ise sırasıyla 207 ve 175 maksimum puan ile bunu takip etmiştir.

Tablo 5.4: Başarıyla tamamlanan bölümlerin tur bazındaki istatistikleri.

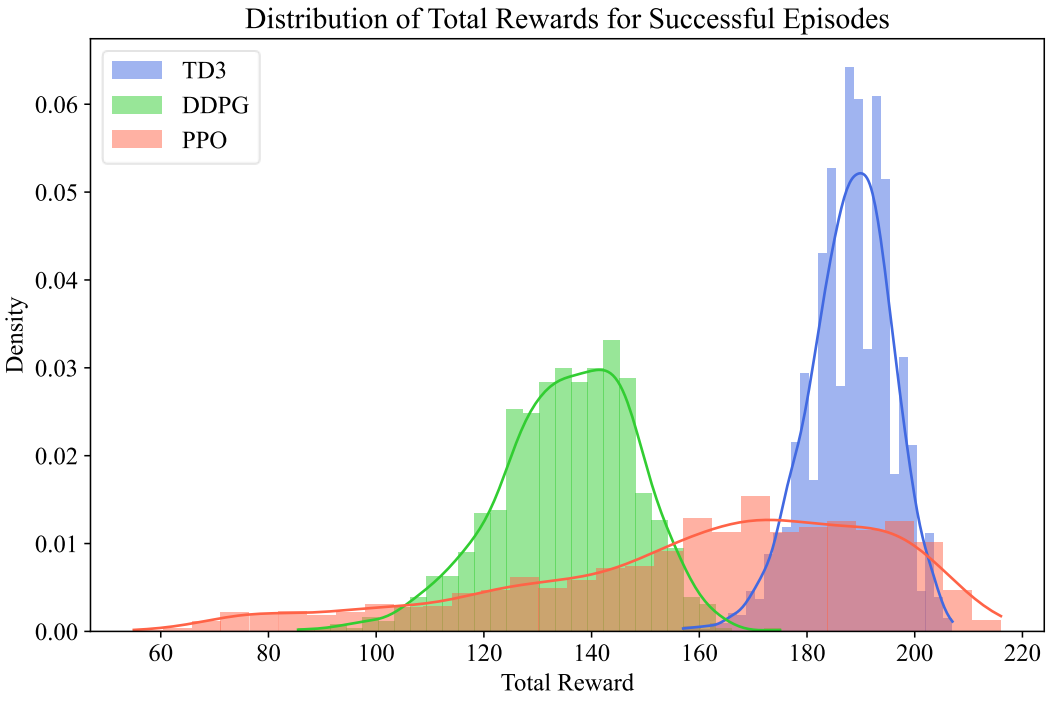
Başarılı turlarda	TD3	DDPG	PPO
Ort. Ödül	187.90±0.33	135.42±0.86	160.06±1.45
Ort. Hız (km/h)	99.58±0.144	87.91±0.47	50.94±0.36
En İyi Tur Süresi	90.3	94.9	135.25
Ort. Süre (s)	94.69±0.15	108.03±0.77	191.16±1.65
Maksimum Ödül	207	175	216

Ortalama hız ise başarılı turları tamamlama süreleri üzerinden elde edilen hızdır. TD3 başarılı turlarda 99.58±0.144 km/h ortalamayla turlayarak en hızlı model olurken, DDPG 87.91±0.47 km/h ile onu izlemiştir. PPO modeli ise 50.94±0.36 km/h ile belirgin şekilde daha yavaş performans sergilemiştir. Bu durum PPO'nun tur tamamlama stratejisinde daha temkinli veya dengeli bir yaklaşım benimsediğini düşündürebilir. En iyi tur süresi eğitim boyunca elde edilen en hızlı tur istatistiğidir. Ortalama süre ise tamamlanan turların ortalamasını belirtir. En iyi tur süresi açısından TD3, 90.3 saniye ile en hızlı tura sahip olurken; DDPG 94.9 saniye ve PPO ise 135.25 saniye ile daha yavaş turlar atmıştır. Ortalama süreler de benzer bir dağılım göstermiş; TD3 tur başına ortalama 94.69 saniyede, DDPG 108.03 saniyede ve PPO ise 191.16 saniyede turları tamamlamıştır.

Bu istatistikler TD3'ün, genel olarak en iyi performansı sergilediğini, en hızlı ve en yüksek ödül skorlarına ulaştığını belirtirken; PPO'nun yüksek ödül potansiyeline rağmen yavaş ve istikrarlı bir strateji izlediğini ortaya koymuştur. DDPG ise ortalama bir performans sergileyerek dengeli bir tutum sergilemiştir.

TD3, DDPG ve PPO algoritmalarının başarılı turlarda elde ettiği ödül dağılımları Şekil 5.4’de görselleştirilmiştir. Mavi renkte temsil edilen TD3 algoritması diğerlerine göre daha yüksek ödül değerlerine ulaşarak, 180 ile 220 toplam ödül değeri arasında yoğun bir dağılım sergilemiştir. Bu TD3'ün başarılı turlarda üstün performans gösterdiğini ve yüksek ödül potansiyeline sahip olduğunu ifade etmektedir. Yeşil renkteki DDPG algoritması ise daha düşük ve dar bir ödül aralığında (100 ile 160

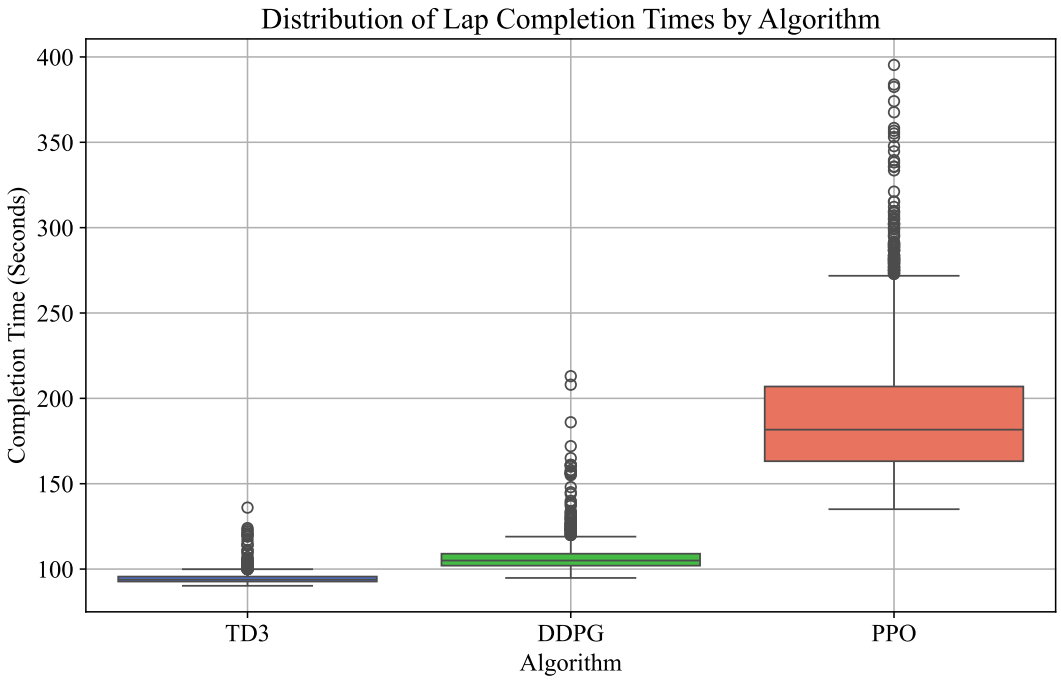
toplam ödül değeri arasında) yer almıştır; bu da algoritmanın başarılı turlarda daha az etkili olduğunu işaret eder. Kırmızı renkteki PPO ise geniş bir dağılım sunmasına rağmen genel olarak daha düşük ödüller elde etmiştir. Bundan dolayı diğer modellere kıyasla tutarsız bir performans sergilediğinden bahsedilebilir. Bu grafik modellerin başarılı turlar sırasında elde ettikleri toplam ödüllerin nasıl dağıldığını detaylı bir şekilde ortaya koyarak, her bir modelin performansını değerlendirme konusunda önemli bulgular sunmaktadır.



Şekil 5.4: Turların başarıyla tamamlandığı bölümlerin ödül dağılımı.

Başarılı bölümlerde elde edilen tur sürelerinin dağılımı Şekil 5.5'te gösterilmektedir. Bu grafik TD3, DDPG ve PPO algoritmalarının tur tamamlama sürelerinin dağılımını karşılaştırmalı olarak sunmaktadır. TD3 algoritması; ortalama tur süresi açısından en iyi performansı göstermiş olup, tur sürelerinin çoğunluğu 100 saniyenin altında gerçekleşmiştir. DDPG algoritması, 150 saniye civarında daha geniş bir tur süresi dağılımına sahiptir ve bazı turları oldukça uzun sürmüştür. PPO ise oldukça geniş bir dağılım ile en uzun tur sürelerine sahip olup, çoğu turu 200 saniye ve üzerinde tamamlanmıştır. Bu görsel farklı algoritmaların tur süreleri açısından

karşılaştırmalı performansını gözler önüne sermekte ve PPO'nun belirgin şekilde daha yavaş olduğunu, TD3'ün ise hız açısından en etkili algoritma olduğunu göstermiştir.



Şekil 5.5: Başarılı bölümlerde elde edilen tur sürelerinin dağılımı.

Bu bölümde son olarak eğitim süresince modellerin yaptığı küçük hata sayıları incelenmiştir. Bu olaylara ilişkin veriler Tablo 5.5'te yer almaktadır. Kaza, aracın yüksek hasar almadığı durumları; pist limiti ihlali, kısa süreli ya da küçük pist dışına taşma durumlarını; yavaş ilerleme, aracın bir süre yavaş hareket ettiği ancak sonradan hızlandığı; spin ise aracın tamamen kontrolünü kaybetmeden küçük spin atma durumlarını ifade etmektedir. Kaza sayılarına bakıldığında eğitim esnasındaki başarılı turlarda DDPG bir kere kaza yaşamış ve bu sonlandırmaya sebep olmamıştır. PPO ve TD3 için araç eğitim esnasındaki başarı ile tamamladığı turlarda herhangi bir hasar almamıştır. Pist limitlerinin ihlalinde TD3 açık ara farkla iyi performans sergilemiştir. DDPG 2583 kere pist limiti ihlali yaparak en yüksek şerit ihlali yapan model olmuştur. PPO ise DDPG'ye nazaran daha az ihlal yapsa da bu konuda TD3'e göre kötü performans sergilemiştir. TD3 ve DDPG nadiren yavaş ilerleme sorunu yaşarken, PPO bu duruma daha sık yakalanmıştır. Bu bir noktada PPO'nun diğer iki modele kıyasla daha dikkatli veya temkinli sürüş stratejisi izlediğini gösterebilir. Üç model de

başarıyla tamamlanan turlar sırasında herhangi bir spin olayı yaşamamıştır. Bu da modellerin dönüş stabilitesi ve kontrol yeteneklerinin etkinliğini gösterir.

Tablo 5.5: Başarıyla tamamlanan bölümlerde (eğitim süresince) sonlandırmaya sebep olmayan küçük hatalar.

Başarılı bölümlerde küçük hatalar	TD3	DDPG	PPO	Toplam
Kaza (küçük hasarlı)	0	1	0	1
Pist Limit İhlali	743	2583	1651	4977
Yavaş İlerleme	3	9	247	259
Spin	0	0	0	0

Özellikle TD3 modelinin diğerlerine göre daha yüksek ödüllere ulaşması ve stabil bir öğrenme eğrisi göstermesi, bu modelin algoritma stabilitesi ve etkinliği açısından öne çıktığını göstermiştir. DDPG modeli ise başlangıçta TD3 ile benzer performans sergilemesine rağmen zamanla geride kalmıştır ve düşük ödül seviyelerinde kalarak belirli durumlarda daha yavaş öğrenme eğilimi göstermiştir. PPO başlangıçta hızlı bir öğrenme ivmesi sergilemiştir fakat öğrenme sürecinde yaşanan dalgalanmalar ödül kazanımında tutarsızlıklara neden olmuştur.

Dört farklı hata olarak belirtilen; kaza, spin, aracın ilerleme katetmemesi ve pist limitlerinin ihlali gibi olayların ısı haritaları, Bölüm 5.2'de detaylı bir şekilde incelenmiştir. Belirtilen hata durumlarına göre modellerin pist üzerinde zorlandıkları konumlar görsel olarak verilmiştir.

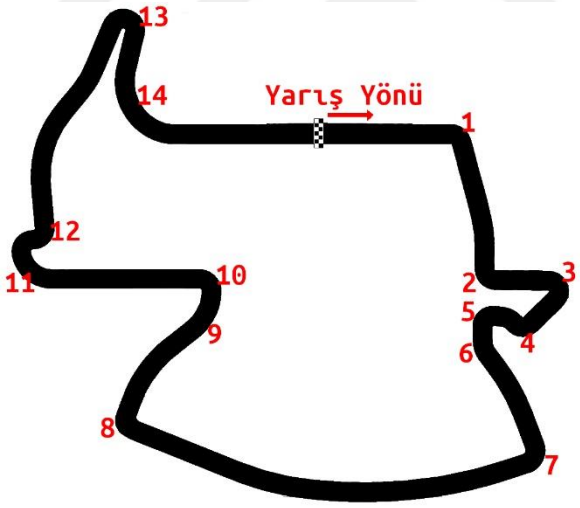
Sonuç olarak elde edilen verilere dayanarak, modellerin eğitim sürecindeki performanslarının yanı sıra modellerin adaptasyon kabiliyetleri ve eğitim esnasında karşılaştıkları zorluklara verdiği reaksiyonlar hakkında önemli bilgiler belirtilmiştir.

5.2 Eğitim Sürecine Ait XAI Sonuçları

Bu alt bölümde eğitim esnasında toplanan verilerden yararlanarak Aalborg pisti üzerinde oluşturulan ısı haritaları üzerinden değerlendirmeler yapılmıştır. Bu haritalar Şekil 5.7, Şekil 5.8, Şekil 5.9 ve Şekil 5.10'da verilmiştir. Isı haritalarındaki renkler ilgili olaya ait yoğunlukları temsil etmektedir; kırmızı renkteki alanlar yoğunluğun çok

olduğu bölgeleri ifade ederken, mavi renkteki alanlar ise düşük yoğunluklu yerleri göstermektedir. Haritaların sağında yer alan bar, yoğunluklara ilişkin renk haritalarını belirtmektedir. Haritalar algoritmaların belirli pist bölümlerindeki performanslarını değerlendirmek için görselleştirme imkânı sunmuştur. Ayrıca bu veriler EK-A ve EK-B'deki tablolarla zaman içerisindeki dağılımlarla birlikte değerlendirilmiştir. Yarış pistinde belirten ısı haritaları, modellerin zayıf yönlerini ve potansiyel iyileştirme alanlarını ortaya koymuştur.

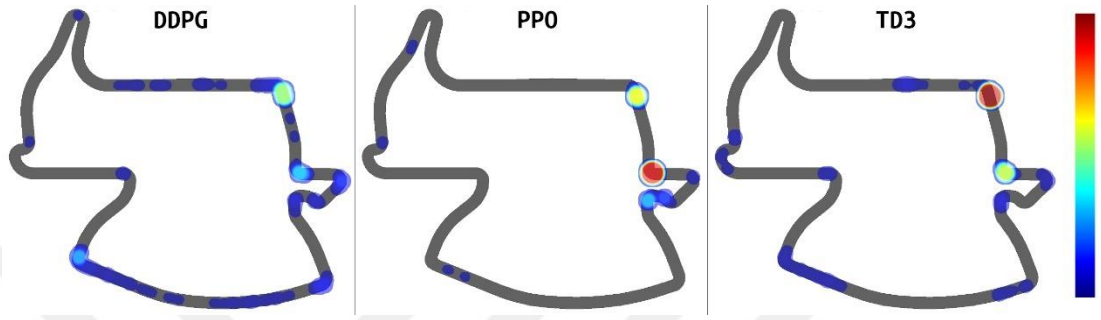
Isı harita yorumlarının daha net anlaşılabilmesi için Şekil 5.6'da Aalborg pistine ait görsele yer verilmiştir. Burada numaralar viraj numaralarını ifade etmekte olup damalı şeridin olduğu bölge yarışın başlangıç noktasını belirtmektedir. Pistin yarış yönü saat yönündedir (şekildeki ok yönünde). Pist toplam 14 viraja sahiptir. Bu görsel pistin yapısal özelliklerini ve hangi bölümlerinin daha teknik ya da hız gerektiren kısımlar olduğunu anlamak için önemlidir.



Şekil 5.6: Aalborg pistinin viraj numaraları.

Şekil 5.7'da gösterilen ısı haritaları, üç algoritmanın 4000 bölümlük eğitim süresince Aalborg pistinde en sık kaza yaptıkları yerleri göstermektedir. Haritalardan elde edilen bilgiler, her algoritmanın belirli virajlar ve pist bölümlerinde nasıl performans gösterdiğini gözler önüne sermiştir. TD3 algoritmasının ısı haritasında; başlangıç çizgisine yakın bölgelerde, ayrıca özellikle 1. ve 2. virajda yüksek kaza yoğunluğu görülmektedir. Bu TD3 algoritmasının eğitim sürecinde bu virajlarda

uygun manevra stratejilerini geliştirmede zorlandığını ifade edebilir. Ancak EK-A'da (bkz. Tablo 8.1) verilen tabloya göre eğitim sürecindeki kaza sayılarının genel olarak düşük olması ve eğitim süreci ilerledikçe kaza sayılarındaki belirgin düşüş göz önüne alındığında, TD3 algoritmasının bu bölgelerdeki performansının zamanla iyileştiği gözlemlenmiştir.



Şekil 5.7: Üç modelin kaza yaptığı bölgelerin pist üzerinde ısı haritaları.

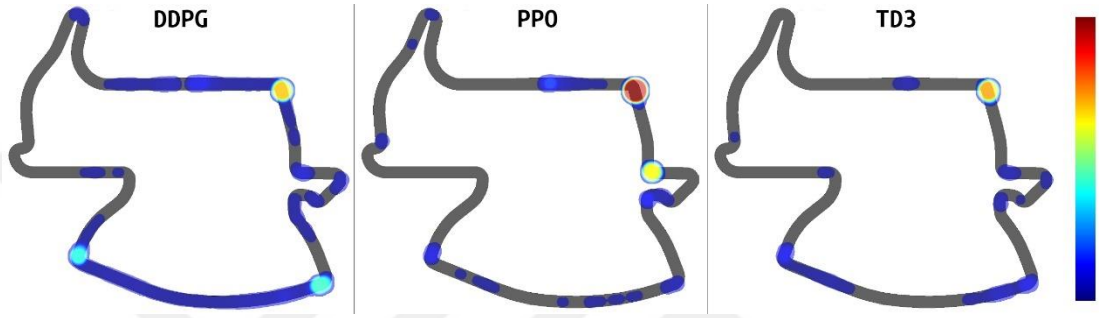
DDPG algoritmasının ısı haritası; başlangıç çizgisinde TD3'e benzerlik göstermiş ve özellikle viraj 1, 2, 3 ve 8'de belirgin kaza yoğunlukları gözlemlenmiştir. DDPG'nin kaza sayılarındaki yüksek yoğunluk, bu virajlarda tutarlı performans göstermekte güçlük çektiğini işaret edebilir. DDPG'nin adaptasyon sürecinde zorlanma gösterdiği ve kaza oranlarının diğer algoritmalara göre daha yüksek olduğu gözlemlenmiştir. Ancak etap sayılarına göre kaza sayıları da değerlendirildiğinde 1. etaptan sonra yarı yarıya azalma görülmüştür.

PPO'nun diğer iki modele göre tüm pist üzerinde daha az kaza yoğunluğuna sahip olduğu görülmektedir. Ancak viraj 1, 2 ve 5 bölgelerinde yoğunlaşmış kazalar görülmektedir. PPO'nun kaza sayılarının özellikle etap 3 ve 4'te düşük olması (bkz. Ek-A, Tablo 8.1) üzerine; algoritmanın eğitim sürecinde belirli zorlukları hızla aşmayı başardığını, adaptasyonunun diğer iki modele göre daha etkili olduğu gözlemlenmiştir.

Üç model değerlendirildiğinde özellikle TD3 algoritması, başlangıçtaki yüksek kaza oranlarını azaltmayı başararak sürekli bir iyileşme göstermiştir ancak PPO kadar düşük kaza sayısına ulaşamamıştır. Buna rağmen hız konusunda PPO'ya kıyasla daha baskın bir performans sergilemesi, TD3'ün daha keşif odaklı bir yaklaşım izlediğini ifade etmektedir. Bu durum TD3'ün risk alarak daha yüksek hızlar elde etme

potansiyeline sahip olduğunu ve bu stratejinin hem avantajlarını hem de dezavantajlarını yansıttığını gösterebilir.

Şekil 5.8, Aalborg pistinde TD3, DDPG ve PPO algoritmalarının eğitim sürecinde sıkça pist sınırlarını ihlal ettikleri bölgeleri ısı haritası ile ifade etmektedir. Renklerin yoğunluğu ihlal olaylarının sıklığını ve kırmızıdan maviye doğru renk değişimi ise daha yüksekten daha düşük ihlale doğru geçişi simgeler.



Şekil 5.8: Üç algoritmaya ait modellerin pist limitlerini ihlal ettiği bölgelerin pist üzerinde ısı haritaları.

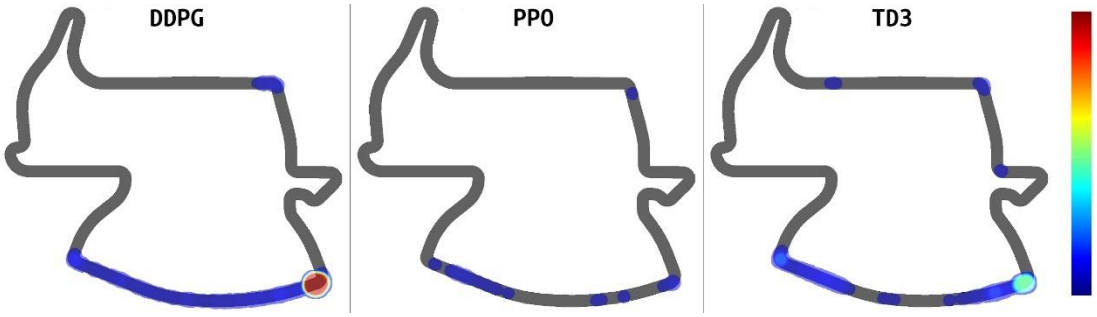
TD3 modeli, 1 numaralı virajlarda yoğun pist sınırı ihlalleri gerçekleştirdiğini göstermektedir. TD3 eğitim sürecinde pist sınırlarına uyum sağlamada gelişme göstermiştir. Ayrıca toplam ihlal sayısı yüksek olsa da (bkz. Ek-A) sonlandırmaya sebep olacak ihlal sayısı (bkz. Ek-B) oldukça düşük seyretmiştir. Bu durum modelin şerit sınırlarını kabul edilebilir ölçülerde ihlal ettiğini göstermektedir.

DDPG modeli 1, 7 ve 8 numaralı virajlar başta olmak üzere pistin büyük kısmında pist limiti ihlallerinde bulunmuştur. Özellikle düzlüklerde diğer algoritmalara göre düşük performans sergilemiştir. Sonuç olarak DDPG'nin eğitim süresince pist sınırlarını sürekli olarak ihlal etme eğiliminde olduğu ve bunun özellikle pistin büyük kısmında belirgin olduğu gözlemlenmiştir.

PPO 1 ve 2 numaralı virajlarda belirgin ölçüde ihlaller yapmıştır. Özellikle ilk virajda diğer iki algoritmaya göre yoğunluğunun yüksek olduğu görülmektedir. Sonlandırma sayıları ve küçük ihlal sayıları (bkz. Ek-A, Ek-B) her bir etap için incelendiğinde ilk virajı diğer algoritmalara göre daha geç öğrendiğinin çıkarımı yapılmıştır. İlk virajın yüksek hızlı düzlük sonrasında sert bir viraj olma karakteristiği

göz önünde bulundurulduğunda zor koşulları diğer algoritmalara göre geç öğrendiği söylenebilir. Birinci virajdan sonraki virajda da ihlal sayısının yüksek olması bunu doğrulamaktadır.

Şekil 5.9'te verilen Aalborg pistindeki spin ve ters yön kazalarını gösteren ısı haritaları, aşama bazında kaza verileriyle birlikte değerlendirilmiştir. TD3 modelinin, çoğunlukla 7 ve 8 numaralı virajlarda yoğunlaşan, orta az yoğunlukta bir spin olayı gerçekleştirdiği görülmektedir. Buna rağmen TD3'ün ısı haritası, nispeten daha az sorunlu alan göstermekte ve DDPG'ye kıyasla pistin zorluklarının daha iyi idare edildiğini göstermektedir. Spin olayları zamana göre incelendiğinde ise ilk etapta TD3 iyi bir performans göstermiştir. Ancak orta aşamalarda muhtemelen keşif (exploration) safhaları nedeniyle spin sayısında artış göstermiştir (bkz. EK-A). Ayrıca bu durum TD3'ün 2. ve 3. etaplarda stratejisini stabilize etmeden önce daha agresif manevralar ve daha yüksek hızlar keşfediyor olabileceğine de işaret etmektedir. TD3 modeli son etaplarda uyum sağlayarak kazaları azaltmış, pozitif yönde genel istikrar ve öğrenme performansı göstermiş; eğitim ilerledikçe daha iyi kontrol ve karar verme becerisine sahip olmuştur. Bu sayede de yüksek hızda agresif manevralar sergileme kabiliyeti ile daha hızlı tur süreleri ve yüksek ödüller elde etmiştir.



Şekil 5.9: Üç modelin spin attığı ya da ters yönde ilerlediği bölgelerin pist üzerinde ısı haritaları.

DDPG modelinin eğitimi esnasında tüm etaplar boyunca sürekli olarak yüksek spin olaylarının (bkz. EK-A) meydana gelmesi, özellikle keskin dönüşlerde ve karmaşık kesimlerdeki pist dinamiklerini idare etmede ve modelin davranışını stabilize etmede zorluk yaşandığını göstermektedir. Model, özellikle en hızlı düzlüklerden birinin ardından gelen 7 ve 8 numaralı virajlarda, bu virajlar arasındaki yüksek hızdaki hafif eğimli düzlükte ve ilk virajda önemli spin aktivitesi yaşamıştır. Bu bölgelerdeki

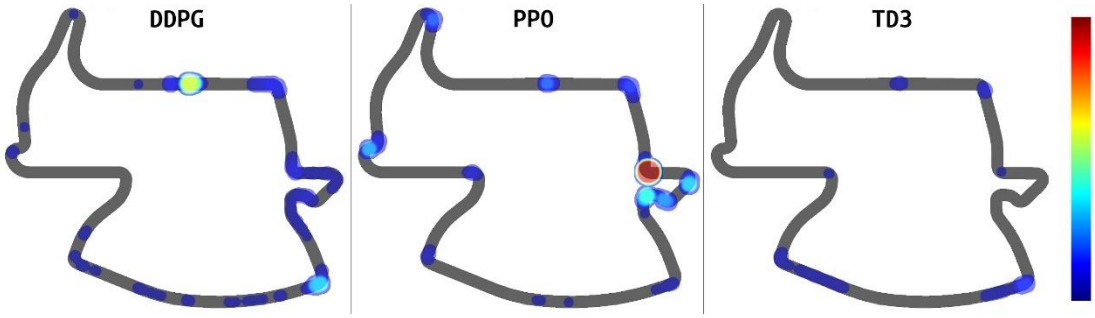
yüksek frekanslı spin olayları modelin ani yön değişikliklerini idare etmekte ve dar dönüşler sırasında dengeyi korumakta zorlandığını göstermiştir.

PPO modeli, ağırlıklı olarak 7 ve 8 numaralı virajların arasında düşük yoğunlukta olmak üzere birkaç spin olayı sergilemiştir. Diğer olayların aksine PPO, TD3 ve DDPG'ye kıyasla daha az spin olayı gerçekleştirmiş; pistin çoğu bölümünde daha iyi stabiliteye sergilemiştir. Ancak diğer parametreler de göz önünde bulundurulduğunda, PPO genel olarak yüksek hızlara çıkmamıştır. Spin olayının yaşanması için aracın yüksek hızlara çıkabilmesi önemli bir etmendir. Bu durum PPO'nun diğer modellere göre daha az spin atmasına neden olmuştur. Ancak PPO'nun düşük hızlarda seyretmesine rağmen spin olayı yaşaması, viraj esnasında yaşanan ani gaz ve direksiyon değişimleri ile açıklanabilir. Ayrıca düşük hız politikası, bu modelin riskten kaçınan, temkinli bir sürüş politikasına sahip olduğunu işaret ediyor olabilir. Özellikle 7 ve 8 numaralı virajlar gibi teknik olarak zorlu bölümlerde PPO'nun gösterdiği performans, bu virajları daha yavaş geçiş stratejisiyle daha kontrollü bir şekilde yönetebildiğine işaret etmektedir. Ancak bu yaklaşım, genel yarış süreleri ve ortalama hız gibi performans metrikleri üzerinde olumsuz bir etki yapabilir. Bu nedenle PPO modelinin sürüş stratejisi ve karar verme süreçleri üzerinde daha detaylı optimizasyonlar yapılması gerekebilir. Modelin bu virajlarda yaşadığı spin olayları, belki de daha agresif bir sürüş tarzı benimsemesi gerektiğini veya belirli viraj teknikleri üzerine daha fazla eğitim yapılmasının yararlı olabileceğini göstermektedir.

Araçların ilerlemediği veya önemli ölçüde yavaşladığı alanlar için ısı haritası Şekil 5.10'da yer almaktadır. Isı haritası ve bu tür olaylara ilişkin aşama bazlı veriler (bkz. EK-A) ile modellerin farklı pist segmentlerinde ve koşullarında değerlendirmesi yapılmıştır.

TD3'e ait ısı haritası, özellikle eğitim ilerledikçe durma veya aşırı yavaşlama vakalarının minimum düzeyde olduğunu göstermektedir. İlk aşamalarda daha yüksek vakalar görülmüş (1. etapta 60 olay), ancak sonraki aşamalarda belirgin bir iyileşme sağlamıştır. Bu durum TD3'ün öğrenme algoritmasının, pistin taleplerine etkin bir şekilde uyum sağladığını ve pistin daha yavaş veya daha karmaşık bölümlerinde yol tutuşu ve karar vermeyi geliştirdiğini göstermektedir. Ayrıca spin olaylarına ait ısı haritasıyla karşılaştırıldığında, en çok spin attığı 7 numaralı virajın çıkışında modelin yavaşladığı görülmüştür. Bu durum modelin spin atmamak için viraj çıkışında daha

temkinli davranarak yavaş hızlanma stratejisini benimsediği şeklinde değerlendirilebilir.



Şekil 5.10: Araçların ilerleme kaydetmediği ya da çok yavaşladığı bölgelerin pist üzerinde ısı haritaları.

DDPG modeli, özellikle sonraki aşamalarda en yüksek sayıda ilerlememe veya yavaşlama örneklerini sergilemiştir. Bu sık yavaşlama, karmaşık pist bölümlerini idare etmede zorluklara veya sık sık durmaya neden olan aşırı temkinli bir politikaya işaret etmektedir. Modelin zorlu pist bölümlerinde daha yavaş bir sürüş politikası sergileyerek ihlallerden kaçındığı da söylenebilir. Bu durumun, keşif ve sömürü (exploration vs. exploitation) arasında daha az optimal bir dengeden kaynaklanabileceği düşünülmüştür.

İlginç bir şekilde, PPO modeli eğitimin orta aşamalarında (özellikle 2. etapta 1316 olay, bkz. EK-A) bu olaylarda bir zirve yapmış ve ardından önemli bir düşüş (4. etapta 172 olay) yaşamıştır. Bu durum modelin başlangıçta belirli pist unsurlarıyla başa çıkmakta zorlandığını ancak eylem politikasını artan deneyim ve hassas karar alma süreçlerine göre güncelledikçe, önemli ölçüde iyileştiği dik bir öğrenme eğrisine işaret ediyor olabilir.

Bu analizler her bir algoritmanın karmaşık pist koşullarında farklı tepkiler verme kapasitesini ve belirli viraj veya bölümlerde ihtiyaç duyulan stratejik değişiklikleri açıkça ortaya koymuştur. Algoritmaların özellikle zorlanılan virajlarda nasıl optimize edilebileceği ve eğitim sürecinde karşılaşılan problemleri aşmak için alınabilecek aksiyonlar konusunda değerli öngörüler sunulmuştur. Bu görsel veriler sayesinde modellerin pist üzerindeki davranışlarını daha iyi anlama ve özellikle zorlandıkları virajları belirleme imkânı elde edilmiştir. Modellerin eğitim sürecindeki

bu tür bilgiler, model dinamiklerini ve sürüş stratejilerini iyileştirmek için önemli yönlendirmeler sunmuştur.

İhlal oranlarının yüksek olduğu virajlarda, modellerin algoritma parametreleri ve ödül mekanizmaları iyileştirilerek genel performansların artırılmasına yardımcı olunabilir. Özellikle TD3, DDPG ve PPO modellerinin performans analizleri, bu modellerin sürüş stratejileri ve karar verme süreçleri hakkında değerli bilgiler sunmuştur. Bu analizler her bir modelin sürüş dinamikleri ve algoritmik karar verme süreçlerinin derinlemesine incelenmesini sağlamış ve modellerin özellikle zorlu virajlar ve hızlı düzlükler gibi çeşitli pist koşullarına nasıl tepki verdikleri konusunda kapsamlı bilgiler sunmuştur.

Bu bulgular her modelin farklı yaklaşımlarının ve Aalborg pistinin ortaya koyduğu yarış dinamiklerine adaptasyon kabiliyetlerinin altını çizmektedir.

5.3 Modellerin Test Yarışındaki Performansına Ait Sonuçları

Önceki bölümlerde modellerin eğitim turlarına ait sonuçlar irdelenmiştir. Bu alt bölümde ise eğitim sonrası elde edilen modeller değerlendirilecektir. Bu amaçla TD3, DDPG ve PPO modelleri kullanılarak Aalborg pistinde gerçekleştirilen 25 turluk yarış; algoritmaların karmaşık ve dinamik sürüş koşullarında karar verme mekanizmalarını değerlendirmek için bir platform sağlamıştır. Tablo 5.6'da yarışa dair toplam süre (yarışı tamamlama süresi), maksimum hız, en hızlı tur süresi (25 tur içerisinde modelin en hızlı olduğu turun süresi), toplam ödül gibi kritik performans metriklerine yer almaktadır.

Yarışın her bir turunda kaydedilen gaz, fren, direksiyon, açı ve hız gibi temel sürüş parametrelerinin analizi, modellerin belirli virajlarda ve düzlüklerde nasıl performans gösterdiklerini detaylı olarak açığa çıkarmaktadır. Bu analizler modeller hakkında değerli öngörüler sağlamıştır.

Toplam süre açısından bakıldığında TD3 modeli 36 dakika 44 saniye ile yarışı en hızlı tamamlayan model olmuştur. DDPG 38 dakika 33 saniye, PPO ise 59 dakika 36 saniyede yarışı tamamlayabilmiştir. Bu TD3 modelinin verimliliğini ve yüksek

performansını vurgulamaktadır. DDPG ve özellikle PPO'nun daha yavaş tamamlama süreleri, bu modellerin karşılaştığı zorlukları ve yarış sırasındaki değişen adaptasyon süreçlerini göz önüne sermiştir.

TD3 hem DDPG hem de PPO'dan daha yüksek performans göstererek düzlükte 190 km/h maksimum hıza erişmiştir. TD3 ayrıca 105.7 km/h ile ortalama hızda da öne çıkmıştır. Dolayısıyla daha agresif ve hız odaklı bir sürüş politikasına işaret etmektedir. Buna ek olarak TD3 en hızlı turu 88.31 saniyede tamamlamış; çevikliğini ve hızlı tepki kabiliyetini sergilemiştir. DDPG ve PPO'nun bu alandaki daha yavaş performansları, bu modellerin performanslarını artırmak için daha fazla optimizasyona ihtiyaç duyabileceğini göstermektedir.

DDPG'nin 9 olay ile en yüksek kaza sayısına sahip olması, daha riskli sürüş stratejilerinin benimsenmiş olabileceğine işaret etmektedir. Buna karşın TD3'ün yarış boyunca hiç kaza yapmaması güvenilirliğini ve etkinliğini göstermektedir. Pist sınırı ihlalleri, yavaş ilerlemeler ve dönüşlere ilişkin istatistikler de benzer şekilde modellerin sürüş kalitesi ve güvenliğine ilişkin kritik bilgiler sağlamıştır. Spinler açısından DDPG 5 olay, PPO 2 farklı olay kaydederken, TD3 hiç spin vakası yaşamamıştır. Spinler, özellikle yüksek hızlarda veya daha zorlu pist bölümlerinde (örneğin sert virajlarda ya da çim alana taşma durumlarında) kontrol kaybını gösterir ayrıca yarış zamanını ciddi şekilde etkileyebilir. TD3'ün hiç spin kaydetmemiş olması, yüksek hızlarda daha kararlı ve güvenilir bir performans sergilediğini bir kez daha gösterirken; DDPG ve PPO'nun spin olayları, yüksek hızlı durumlar veya karmaşık manevralar altında istikrar ve kontrolde sorunlar yaşadıklarına dair ipuçları sunmuştur.

Tablo 5.6: Üç modelin yarıştığı 25 Turluk test yarışına ait istatistikler.

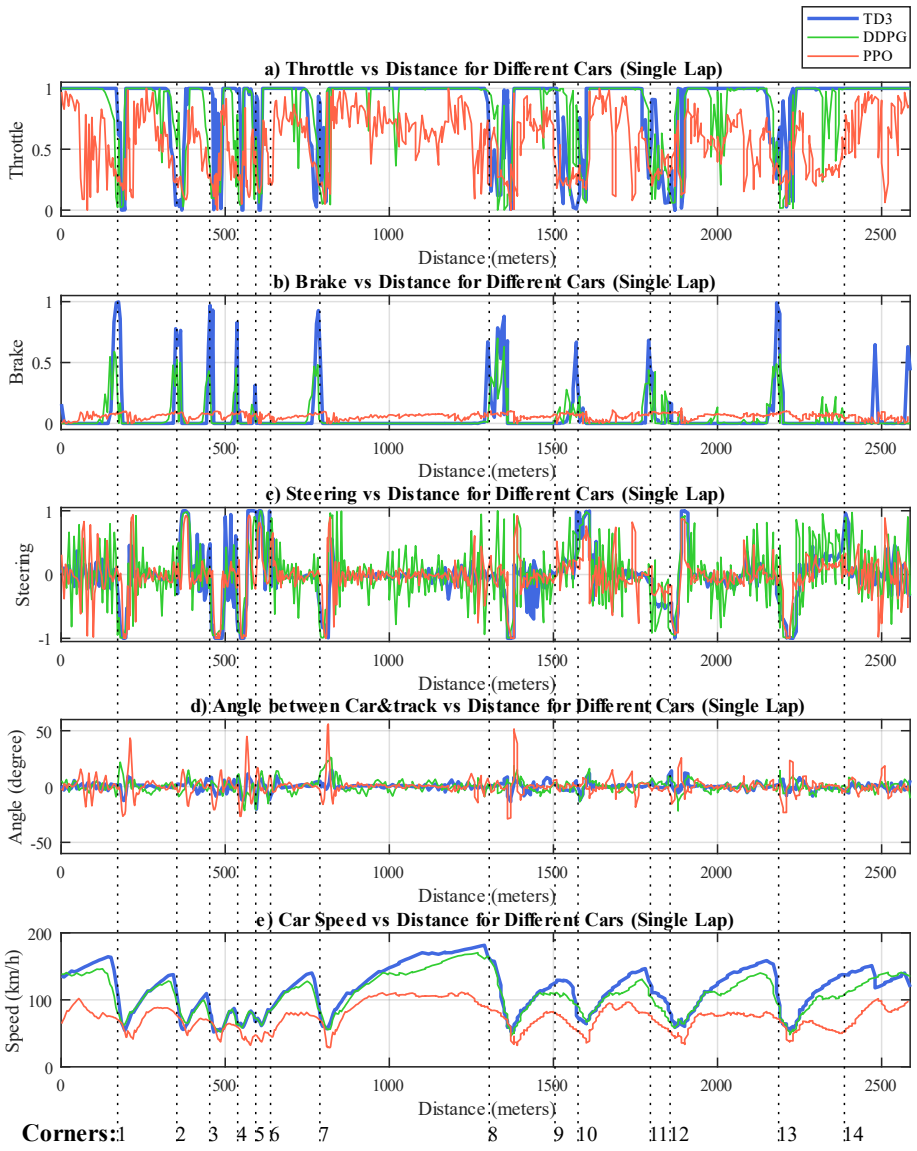
25 Turluk Test Yarışında	**TD3**	**DDPG**	**PPO**
Toplam Süre (dk/s/ms)	**36:44.120**	38:32.972	59:36.32
Azami Hız (km/h)	**190**	174	112
En Hızlı Tur (s)	**88.31**	90.298	140.776
Toplam Ödül	**13863**	4498	5283
Ort. Hız (km/h)	**105,7**	100,7	65,1
En Hızlı Turdaki Ortalama Hız (km/h)	**105,5**	103,1	66,2
Kaza Sayısı	**0**	9	1
Pist Limiti İhlali sayı	**13**	111	25
Yavaş İlerleme sayısı	3	37	**1**
Spin Atma sayısı	**0**	5	2
Yarış Mesafesi (m)	64.709		

Şekil 5.11'de TD3 (mavi), DDPG (yeşil), ve PPO (turuncu) modellerinin 25 turluk yarış sırasında en hızlı turuna ait temel sürüş parametreleri (gaz, fren, direksiyon, hız ve araç ile yol arasındaki açı) konum bilgisine (distance, aracın başlangıç noktasına olan uzaklığı) ve viraj numaralarına göre gösterilmiştir. Bu grafikler; her bir modelin farklı pist noktalarındaki performansını, özellikle her virajda nasıl manevra yaptıklarını, gaz ve fren kullanımlarını ve hız değişikliklerini detaylı bir şekilde analiz etmek için kullanılmıştır.

Grafikler TD3 modelinin genel olarak daha stabil ve sürekli bir gaz kullanım politikasına sahip olduğunu gösterirken, DDPG ve PPO modelleri daha değişken bir gaz kullanımı politikası sergilemiştir. Frenlemede, özellikle keskin virajlar öncesi DDPG ve PPO'nun daha sık fren yaptığı görülmüştür. TD3'ün frenleme frekansının daha düşük ve daha düzgün olduğu görülmektedir.

Direksiyon açıları; TD3'ün daha düzgün ve kademeli direksiyon hareketlerine sahip olduğunu, DDPG ve PPO'nun ise daha ani ve sık direksiyon değişiklikleri yaptığını göstermiştir. Bu durum, TD3'ün daha iyi bir yol tutuşu ve istikrarlı sürüş performansına işaret etmiştir.

Hız profilleri, TD3'ün virajlardan çıkışta daha hızlı olduğunu ve düzlüklerde maksimum hıza ulaştığını göstermiştir. Açı grafiği, araçların pist eksenine göre yönlendirmesini ve viraj alma tekniklerini yansıtmaktadır; burada da TD3 modeli genellikle daha az sapma ile daha tutarlı bir yol izlemiştir. DDPG, TD3'e yakın bir sonuç elde etmiş olsa da PPO modelinde sapmaların daha fazla olduğu gözlemlenmiştir.



Şekil 5.11: TD3, DDPG ve PPO algoritmalarının test yarışında en hızlı turlarına ait eylem, hız, açı grafikleri.

Bu analizler her bir modelin pist üzerindeki performansını ve sürüş stratejilerini anlamada, ayrıca model parametrelerinin ve algoritmalarının nasıl optimize edilebileceği konusunda değerli öngörüler sağlamaktadır. Modellerin pist üzerindeki zorlandığı noktalar bu grafiklerle tespit edilebilir. Böylece daha etkin eğitim ve iyileştirme stratejileri geliştirilebilir.

5.4 Eğitilmiş Modellerin SHAP Yöntemine göre Açıklanabilirlik Sonuçları

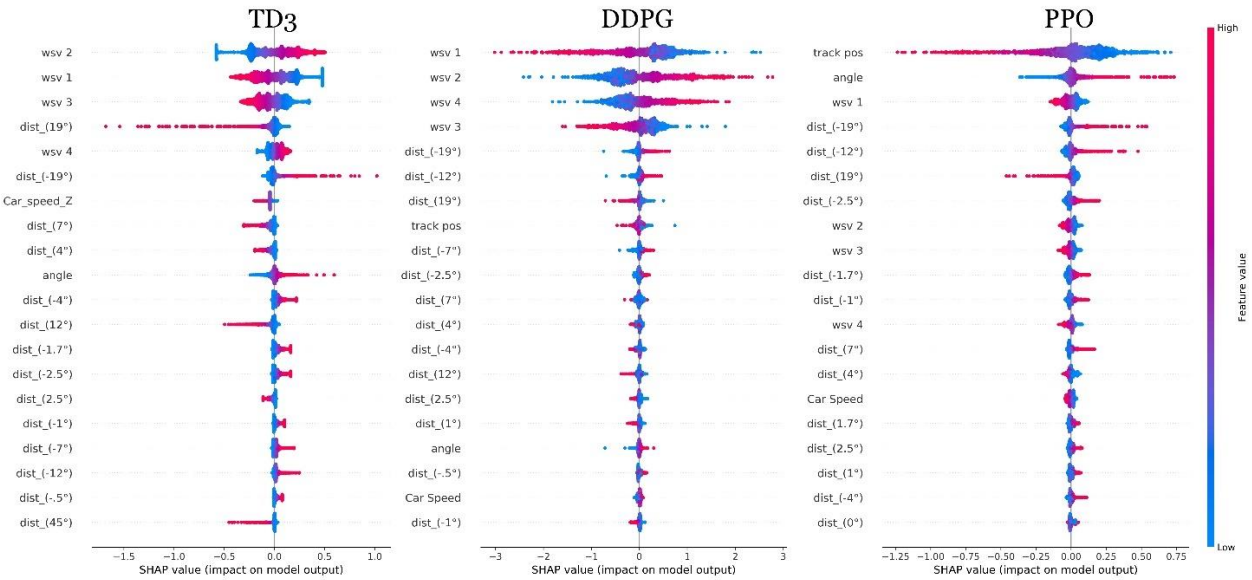
Önceki bölümlerde XAI tekniklerinin teorik temellerini ve önemini ele aldıktan sonra, bu alt bölümde kavramların otonom sürüş üzerinde uygulanabilirliğini somut bir test durumu üzerinden detaylı bir şekilde incelemeye yönelinmiştir. Öncelikle tüm veriler üzerinde analizler yapılarak üç model için global açıklanabilirlik sonuçları verilmiştir. Ardından en başarılı model olan TD3 için çeşitli durumlardaki eylemlerine ilişkin lokal açıklanabilirlik grafikleri analiz edilmiştir.

Bu bölümde grafiklerde yer alan terimlerin açıklamaları şu şekildedir:

- **Car Speed :** Araç hızını (km/h) belirtir. 300'e bölünerek normalize edilmiştir.
- **dist_(açı°) :** İlgili açıya ait sensörün okuduğu uzaklık (m) değeridir. 200'e bölünerek normalize edilmiştir.
- **wsv :** İlgili tekere ait (4 farklı teker için; 1: sağ ön, 2: sol ön, 3: sağ arka, 4: sol arka teker) dönme hızını (rad/s) belirtir.
- **Engine RPM :** Motorun dakikadaki devir sayısı. 10,000 ile normalize edilmiştir.
- **Car speed y ve z:** Aracın yanal ve dikey hızı. 300 ile normalize edilmiştir.
- **angle :** Aracın yol ile arasındaki açı. π ile normalize edilmiştir.
- **track pos :** Aracın orta şeride göre konumu.
- **Throttle :** Gaz.
- **Steering :** Direksiyon.
- **Brake :** Fren.

Şekil 5.12'deki SHAP özet grafikleri, üç farklı DRL modelinin direksiyon çıktısı üzerindeki özelliklerin etkisini göstermektedir. SHAP özet grafiklerinde x-ekseninin sağ tarafı (pozitif kısmı) model çıktısına olan pozitif etkiyi, sol tarafı (sıfırdan küçük) ise negatif etkiyi ifade eder. Modellere uygulanan özellikler (girdiler)

yüksek değere sahip ise kırmızı renkte, düşük değere sahipse mavi renktedir. Her nokta, bir özellik ve belirli bir tahmin için bir SHAP değerini temsil etmektedir.



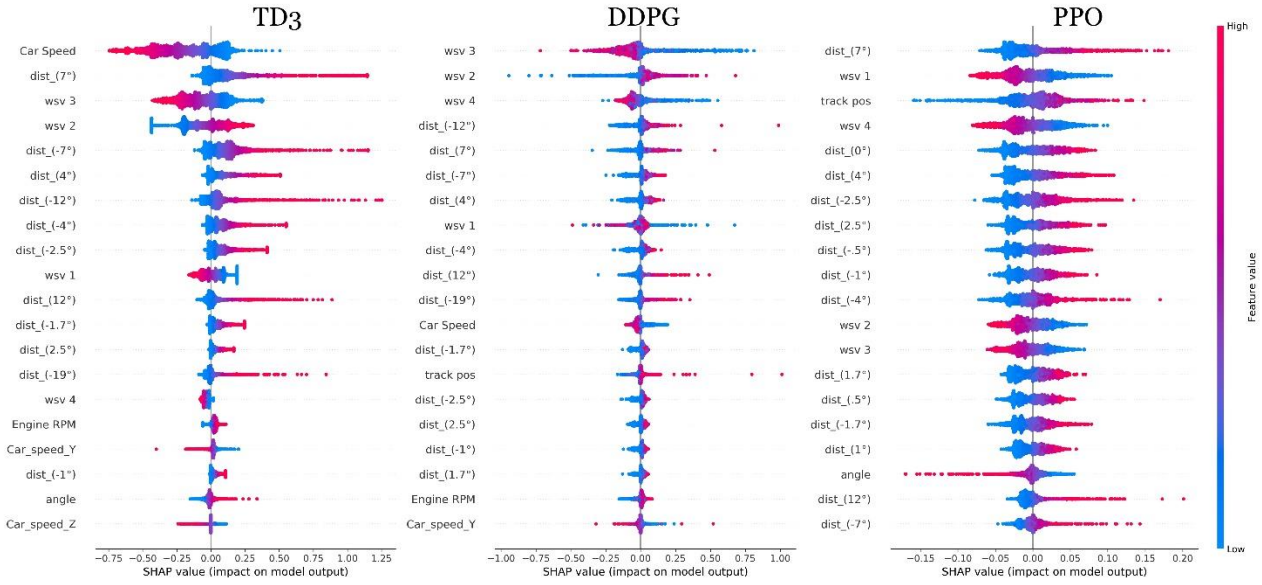
Şekil 5.12: Direksiyon açısı çıktılarına ait SHAP değerlerinin özet grafiği.

TD3 için tekerlek dönüş hızları en önemli etkiye sahiptir. Modelin direksiyon çıktısına 19° ve −19°'deki uzaklık değerleri, sağa ve sola dönüşlerde etkili olmuştur. Bu da aracın şeride yakın olduğu bölgelerde önemli ölçüde etkilendiğini göstermektedir. Ek olarak açı bilgisi de doğrusal bir şekilde direksiyon çıktısına etkili olmuştur. PPO en çok şerit içerisindeki konumundan ve açı değerinden etkilenmiştir. Bu da modelin şeridi ortalamaya çalıştığını, ihlallerden ve kazalardan kaçındığını gösterebilir. DDPG ise TD3'e benzer şekilde dört tekerin dönüş hızından ve 19°'deki uzaklık bilgilerinden etkilenmiştir. Ancak bu değerlerin dağılımı TD3 kadar düzgün değildir.

TD3 ve DDPG modelleri, değişen pist koşullarına daha uyumlu bir strateji benimseyerek daha dengeli kararlar alırken; PPO modeli daha basit bir yaklaşımla pist merkezinden sapmalara göre direksiyon kontrolünü düzenlemiştir. Bu durum TD3 ve DDPG modellerinin en iyi performansı sergilemesinin nedenlerinden biri olabilir.

Şekil 5.13'te gaz pedalı çıktılarına ilişkin SHAP değerlerinin özet grafikleri yer almaktadır. TD3 için gaz çıktısı üzerinde araç hızı en büyük etkiye sahiptir. Yüksek

hızlar (kırmızı) negatif, düşük hızlar (mavi) pozitif etki yaratmıştır. 7° ve −7°'deki uzaklık bilgileri belirgin etkilere sahiptir. Tekerlek hızları da önemli bir faktör olarak göze çarpmaktadır.

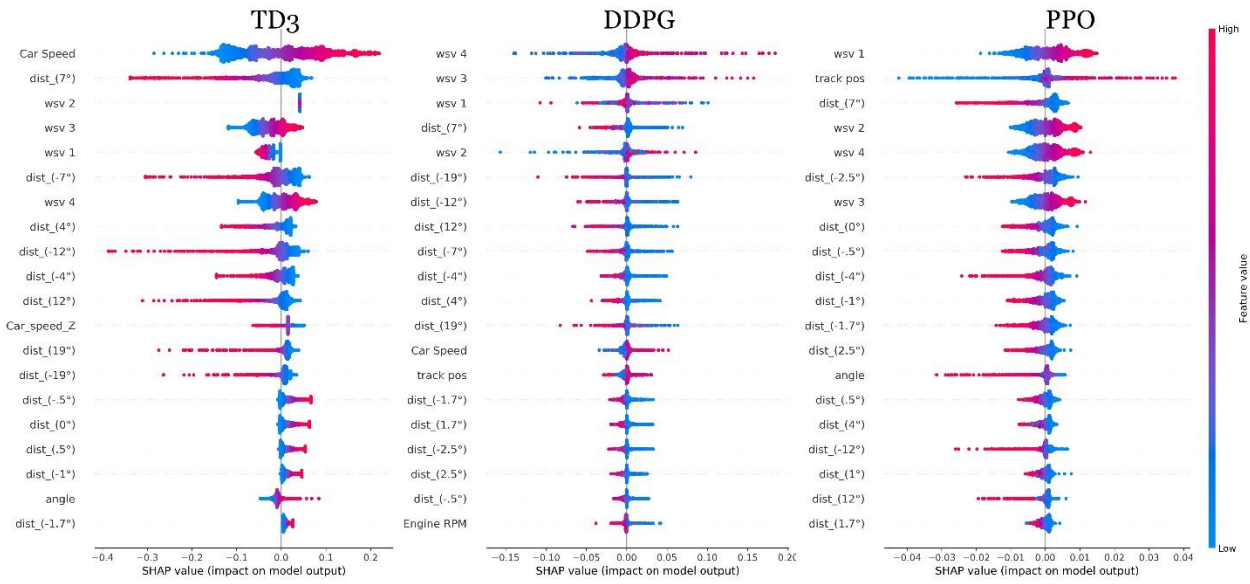


Şekil 5.13: Gaz pedalı çıktılarına göre modellerin SHAP değeri özet grafikleri.

DDPG modelinde tekerlek hızları model çıktısında en büyük katkılara sahiptir. Ayrıca −12°, −7° ve 7° açılarındaki sensörlerin gaz çıktısına belirgin etkileri vardır. PPO grafiğinde sensör mesafeleri gaz çıktısı üzerinde en büyük etkiye sahipken tekerlek hızları da önemli rol oynamaktadır. Ayrıca aracın pist üzerindeki konumu da gaz çıktısını etkileyen faktörlerdendir. Araç hızı TD3 modelinde en belirgin özellik olarak öne çıkmış, PPO ve DDPG modellerinde ise nispeten daha az etkiye sahip olduğu görülmüştür. Özellikle TD3'ün yüksek hızlarda iken gaz çıktısına negatif etki yaptığı görülmektedir. Bu da modelin ivmelenme yönetimi hakkında bilgi vermektedir. Her üç model sensörlerden gelen mesafe bilgilerini dikkate alırken, PPO modelinde daha geniş bir açı aralığında etkili olduğu gözlemlenmiştir. Tekerlek hızları TD3 ve DDPG modellerinde daha belirgin etki göstermiştir. TD3 ve DDPG modellerinin spin sayıları ve hızları eğitimin ilk aşamalarında yüksek olduğu için özellikle TD3 çıktısında yüksek teker hızından negatif etkilenmesi hakkında ipucu vermektedir. Pist pozisyonundan en çok etkilenen model de PPO olmuştur.

Özetle TD3 modeli araç hızına daha fazla ağırlık verirken, PPO modeli sensör mesafelerine ve pist pozisyonuna odaklanmıştır. DDPG modeli ise tekerlek hızlarına ve belirli sensör mesafelerine dikkat etmiştir. Bu farklı stratejiler, modellerin gaz kontrolü üzerindeki karar verme süreçlerinde farklılıklar yaratır.

Şekil 5.14'te frenleme eylemlerine ilişkin özet SHAP grafikleri verilmiştir. TD3'ün frenleme eylemlerinde en büyük etkenlerden birinin araç hızı değerinin yüksek olması ve 7°'deki uzaklık bilgisinin düşük olması etkili olmuştur. Ayrıca $-7°$, $\pm 4°$ ve $\pm 12°$'deki uzaklıkların düşük olması, frenleme çıktısında etkin ve belirleyici olmuştur. Teker hızlarında da benzer durum söz konusudur. DDPG'de teker dönüş hızları, $-19°$ ve $\pm 12°$ açılarındaki mesafeler fren üzerinde belirgin etkilere sahiptir. Araç hızı fren kontrolünde dikkate değer bir faktördür ancak TD3 modeline göre daha az etkilidir. PPO'da tekerlek hızları frenleme çıktısı üzerinde önemli bir rol oynamıştır. Aracın pistin orta şeridine göre konumu frenleme çıktısını doğrudan etkilemiştir. 7°, $-2.5°$ ve 0° açılarının yanı sıra diğer yakın açıdaki mesafeler fren üzerinde belirgin etkilere sahiptir.

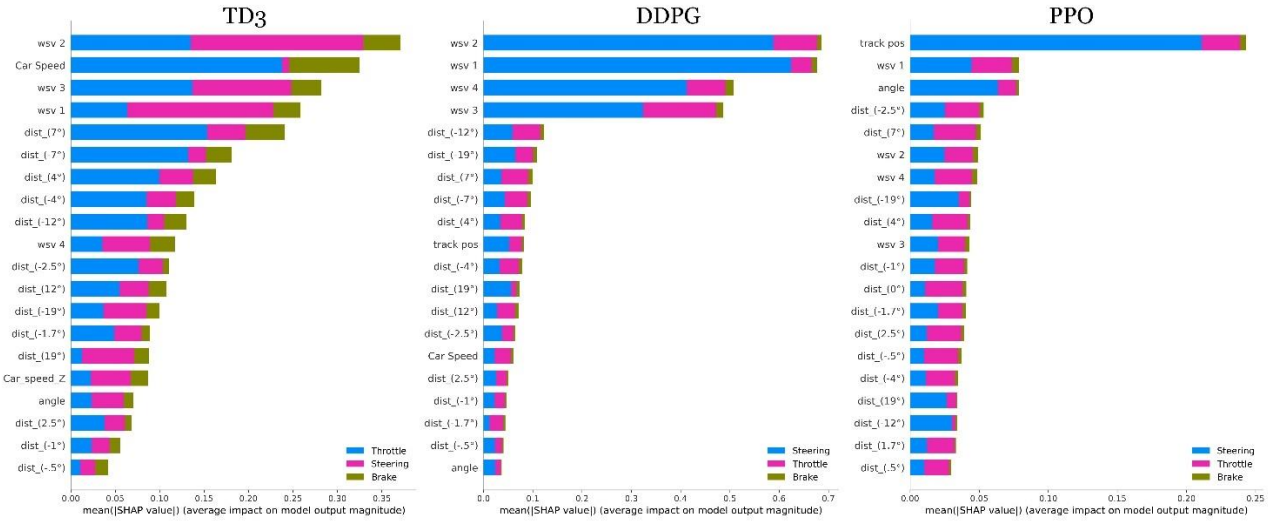


Şekil 5.14: Fren çıktılarına ait SHAP değerleri için özet SHAP grafikleri.

Araç hızı TD3 modelinde en belirgin özellik olarak öne çıkar, DDPG ve PPO modellerinde ise nispeten daha az etkiye sahiptir. Her üç model, sensör mesafelerini dikkate almıştır; ancak TD3 ve DDPG modellerinde daha belirgin etkiler gözlemlenir.

Tekerlek hızları tüm modellerde önemli bir rol oynamıştır ancak DDPG modelinde etkisi daha belirgin olmuştur. Orta şeride göre aracın konumu PPO modelinde önemli bir faktördür, TD3 ve DDPG modellerinde nispeten daha az etkilidir. Elde edilen sonuçlar dikkate alındığında: TD3 modeli araç hızına daha fazla ağırlık verirken, DDPG modeli tekerlek hızlarına ve belirli sensör mesafelerine odaklanır. PPO modeli ise tekerlek hızları ve pist pozisyonuna dikkat etmektedir.

Şekil 5.15, üç DRL modelinin özelliklerin (girdilerin) üç ana eylem üzerindeki çıktılarının etkilerinin genliğinin ortalamasını gösterir. Grafikte her bir özellik için ortalama SHAP değerleri (model çıktısına olan etkinin ortalama büyüklüğü) verilmiştir. Üç modelde gaz ve direksiyon eylemleri üzerinde teker hızları önemli bir etkiye sahiptir. TD3 için araç hızı özelliği gaz ve fren üzerinde önemli rol oynarken, direksiyon açısının çıktısına çok az etkide bulunmuştur. Yüksek hızda ani direksiyon manevraları olumsuz sonuçlara yol açabileceği için burada tutarlı bir davranış sergilediğinden söz edilebilir. Aracın yüksek hızlarda seyir halinde iken viraja giriş ve çıkışlarında dış teker (virajın yönünün tersindeki teker) yönlendirmede daha etkin bir rol oynar. Aracın hareket ettiği pist yüksek hızlarda sağa dönüş içeren virajlardan meydana geldiği için TD3 grafiğinde belirtilen “wsv 2” (sol ön tekerin dönüş hızı) parametresi direksiyon eyleminde belirleyici olmuştur. Ayrıca uzaklık bilgilerinden $7°$ ve $-7°$ başta olmak üzere, diğer uzaklık bilgileri de ivmelenmede etkili olmuştur. DDPG’nin gaz ve direksiyon çıktılarında $-12°$, $-19°$ ve diğer açılardaki uzaklık özellikleri de rol oynamıştır.

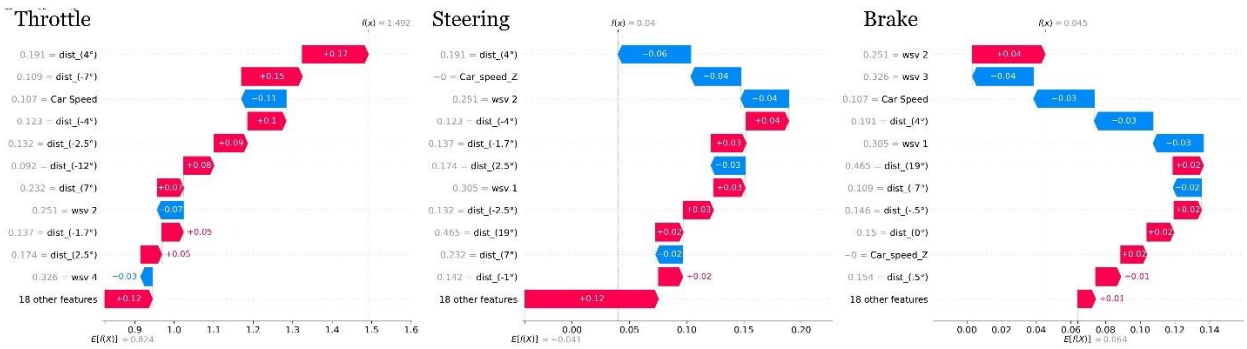


Şekil 5.15: SHAP özet grafikleri.

PPO modelinde ise aracın orta şeride göre konumu direksiyon çıktısında büyük ölçüde belirleyicidir. Bu PPO'nun şerit ortalama stratejisini benimsediğini bir kez daha göstermiştir. Ek olarak $-2.5°$, $0°$, $7°$ ve diğer açılardaki mesafelerin gaz ve direksiyon üzerinde etkileri mevcuttur. Sonuç olarak sensör mesafeleri her üç modelde de dikkate alınır, ancak PPO'da direksiyon üzerinde daha belirgin etkilere sahiptir. Pist pozisyonu PPO'da direksiyon eylemi üzerinde en büyük etkiye sahipken, TD3 ve DDPG'de daha az etki gözlemlenmiştir. TD3 ve DDPG tekerlek hızlarına ve araç hızına odaklanırken, PPO modeli pist pozisyonu ve sensör mesafelerine önem vermiştir; bu da kontrol stratejilerini farklılaştırmıştır.

Global açıklamalar bir modelin tüm durumlar çerçevesinde genel politikaları hakkında bilgi verebilir. Ancak modelin kararlarını lokal olarak incelemek eylem politikaları hakkında daha fazla bilgi sunabilir. Bu nedenle TD3 modeline ilişkin farklı senaryolar değerlendirilmiş ve kararları irdelenmiştir. Böylelikle lokal açıklanabilirlik sunulması hedeflenmiştir. Bu grafiklerde SHAP yöntemine ait Şelale Grafikleri kullanılmıştır. Grafiklerde üç eylem için özelliklerin katkısı incelenmiştir.

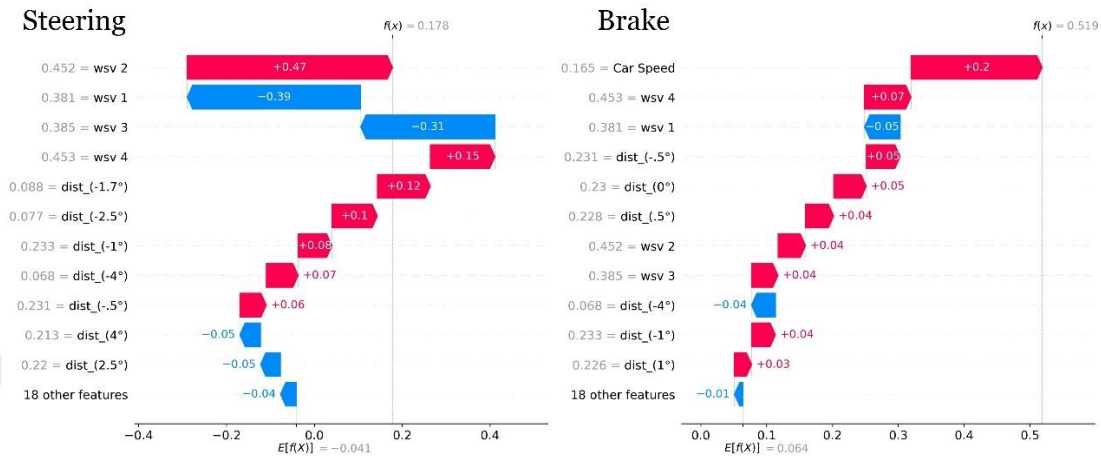
Şekil 5.16'da, model tam gaz eyleminde bulunmuş direksiyon yaklaşık sıfır konumundadır. Sisteme herhangi bir fren eylemi uygulamamaktadır. İvmelenme eylemine en çok $4°$, $-7°$, $-4°$, $-2.5°$'ye ait uzaklık sensörleri pozitif etki yaratmıştır. Direksiyon ve fren eylemi için özellikler çıktıların sıfıra yakın bir değer üretmesini sağlamıştır. Araç düzlük konumunda standart hızlanma eyleminde bulunmuştur.



Şekil 5.16: TD3 modelinin SHAP yöntemiyle bir duruma ait lokal açıklanabilirlik grafiği, örnek:1.

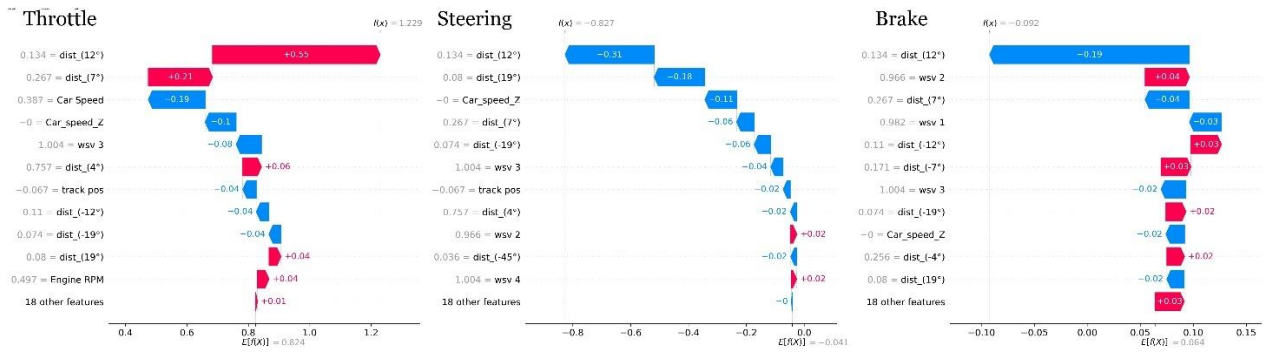
Şekil 5.17'de, araç sol şeride doğru yaklaşmaktadır. Uzaklık sensörlerinden $-1.7°$, $-2.5°$, $-1°$, $-4°$ gibi sensörler aracın direksiyonu sağa doğru çevirmesi

eyleminde etkili olmuştur. Ayrıca viraj esnasında dış teker (sol teker) dönüş hızı olan "wsv 2" parametresi, aracın yönlendirilmesinde en büyük etkiyi sağlamıştır. Aracın frenleme eyleminde en büyük pozitif etkiyi araç hızı yapmıştır.



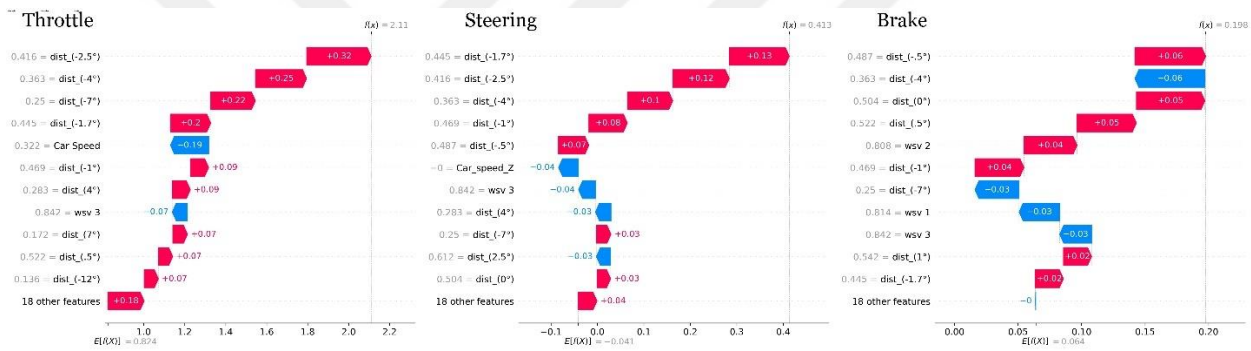
Şekil 5.17: TD3 modelinin SHAP yöntemiyle bir duruma ait lokal açıklanabilirlik grafiği, örnek:2.

Şekil 5.18'de, araç tam gaz eyleminde bulunarak direksiyonu neredeyse tam tur sola çevirmiştir. Frenleme eyleminde bulunmamıştır. Uzaklık sensörlerinden 12° ve 7°'ye ait mesafe değerleri gaz parametresinde etkin olmuştur. Araç hızı ise bir miktar bu eylemi baskılamıştır. Direksiyonun sola çevrilmesinde uzaklık sensörlerinden 12°, 19°, 7°'deki değerler etkili olmuştur. Bu sensörlere bakıldığında aracın sağ şeride konumlanmasının yakın olduğu gözlemlenmiştir. Frenleme eyleminde 12°'deki sensör verisi negatif yönde baskılayarak eylemde bulunmamasını sağlamıştır.



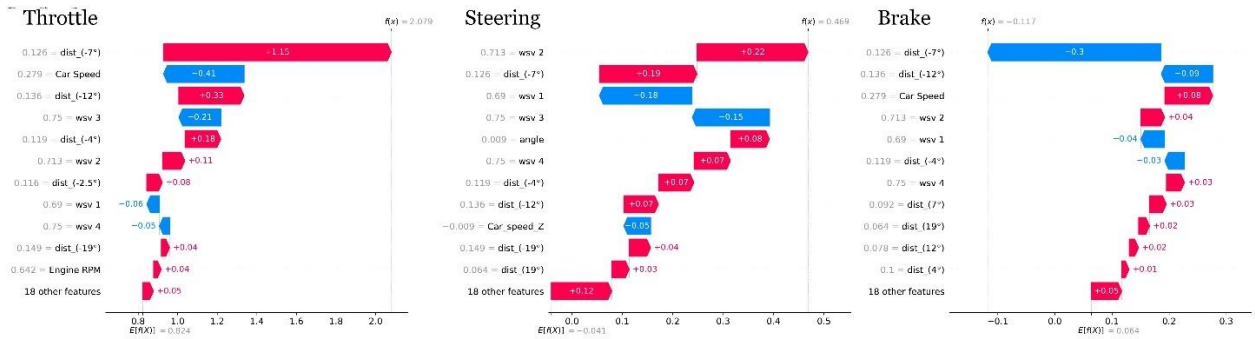
Şekil 5.18: TD3 modelinin SHAP yöntemiyle bir duruma ait lokal açıklanabilirlik grafiği, örnek:3.

Şekil 5.19’da, araç orta-yüksek hızda ilerlerken hafif frenleme yapmıştır. Aracın şeritlere olan mesafesi uzaktır ancak $-0.5°$, $0°$ ve $0.5°$’deki mesafe bilgileri bu eylemin uygulanmasında büyük etki sağlamıştır. Araç ile dikey konumundaki pist sınırına yaklaşık 100 metre uzaklıktayken, hafif frenleme eyleminde bularak aracın hızını viraj girişine yaklaşırken düşürmeyi hedeflediği söylenebilir. Diğer durumlarda daha geniş açı verilerine odaklanırken, burada daha dar açı değerlerindeki mesafelere odaklanması da bunu doğrulayabilir. Direksiyon eyleminde sağa doğru döndürmede etkili olan negatif değerleri açılara (aracın sol kısmı) ait mesafeler, pozitif değerli açılara (aracın sağ kısmı) ait mesafelere göre daha büyüktür. Buna rağmen aracı sağa doğru konumlandırarak virajın girişinde aracı ideal çizgide hareketini sağladığından söz edilebilir.



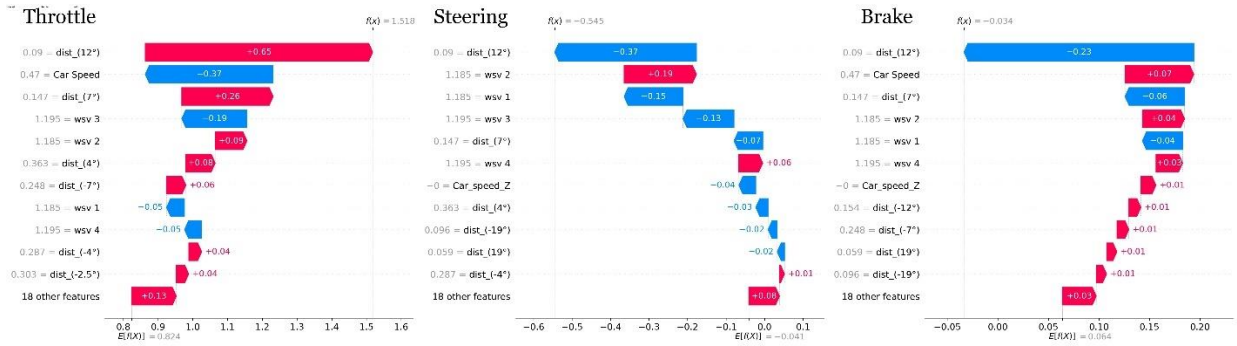
Şekil 5.19: TD3 modelinin SHAP yöntemiyle bir duruma ait lokal açıklanabilirlik grafiği, örnek:4.

Şekil 5.20’de, araç yaklaşık 83 km/h hız ile ilerlerken tam gaz eyleminde bulunarak direksiyonu yarım tur çevirdiği konumda tutmuştur. $-7°$’deki mesafe sensöründen gelen bilgi doğrultusunda aracın sol şeride yakın konumlandığı görülmektedir. Ancak bu durumda frenleme yapmamış hatta bu sensör değerine rağmen frenleme eylemine negatif katkıda bulunmuştur. Buna ek olarak aracın hızlanması için en büyük etkiyi göstermiş, sol ön teker hızı ile direksiyonun sağa doğru yönlendirmesine etki ederek aracı sol şeritten uzaklaştırmaya çalıştığı söylenebilir.



Şekil 5.20: TD3 modelinin SHAP yöntemiyle bir duruma ait lokal açıklanabilirlik grafiği, örnek:5.

Şekil 5.21'de, bir önceki duruma benzer olay daha söz konusudur. Burada araç direksiyon eyleminde yarım tur sola çevirmiştir. Araç 141 km/h hız ile seyrederken, 12°'deki uzaklık bilgisi; tam gaz eyleme devam etmesi, direksiyonu sola çevrilmesi ve frenleme yapmaması yönünde etkide bulunmuştur. Bu durum değerlendirildiğinde, eğer araç frenleme eyleminde bulunsaydı yüksek hızda hareket ettiği için tekerleklerde kayma yaşayarak kontrolü kaybedebileceği bir senaryo oluşabilirdi. Belirtilen durumlar göz önünde bulundurulduğunda gerçekleştirilen eylemlerin tutarlı olduğu söylenebilir.



Şekil 5.21: TD3 modelinin SHAP yöntemiyle bir duruma ait lokal açıklanabilirlik grafiği, örnek:6.

Bu alt bölümde üç farklı DRL modelinin açıklanabilirliğini SHAP yöntemleri kullanarak global ve lokal açıklamalar olarak değerlendirmiştir. Yapılan analizler modellerin direksiyon, gaz ve frenleme eylemleri üzerinde hangi özelliklerin baskın olduğunu göstermiş ve her bir modelin bu eylemler üzerindeki karar verme süreçlerindeki farklılıklar ortaya konmuştur. SHAP özet grafikleri, her bir özelliğin model çıktısı üzerindeki etkisini göstermiş ve bu etkilerin ne kadar büyük olduğu

değerlendirilmiştir. Lokal açıklamalar, TD3 modelinin çeşitli senaryolardaki karar verme süreçlerini detaylı bir şekilde analiz etmiştir. Örnek senaryolar üzerinden yapılan incelemelerde, TD3 modelinin sensörlerden gelen mesafe bilgilerine ve tekerlek hızlarına dayalı olarak gaz, fren ve direksiyon eylemlerini nasıl belirlediği ortaya konmuştur.

TD3 modeli genel olarak araç hızına, DDPG modeli tekerlek hızlarına ve belirli sensör mesafelerine, PPO ise pist pozisyonuna ve sensör mesafelerine duyarlılık göstermiştir. Genel olarak TD3 ve DDPG modelleri daha hızlı bir sürüş stratejisi benimserken, PPO modeli ise pist pozisyonuna ve sensör mesafelerine odaklanarak daha basit ve dengeli şerit ortalama stratejisini benimsemiştir.

TD3 modeline ait çeşitli durumlardaki kararları incelendiğinde; modelin, yüksek hızda direksiyon manevralarını dikkatli bir şekilde yöneterek aracın stabilitesini koruduğu, belirli açılardaki mesafe verilerine göre fren ve gaz eylemlerini optimize ettiği gözlemlenmiştir. Araç hızının yüksek olduğu durumlarda, frenleme eylemlerinde sensör mesafelerinin ve tekerlek hızlarının kritik rol oynadığı, bu verilerin modelin kararlarını nasıl şekillendirdiği açıklanmıştır. Ayrıca viraj giriş ve çıkışlarında, dış teker hızlarının ve mesafe sensörlerinin direksiyon kontrolünde belirleyici olduğu vurgulanmıştır.

5.5 Bölüm Değerlendirmesi

Bu bölüm, DRL tabanlı otonom araçların gerçek dünya senaryolarına adaptasyon kabiliyetini değerlendirirken, bu adaptasyonun öğrenme süreçlerini ve karşılaşılan zorlukları anlamak için önemli veriler sunmaktadır. Algoritmaların performansları, elde ettikleri ödül miktarları ve başarı oranları gibi metriklerle kapsamlı bir şekilde analiz edilmiş. Eğitim sürecinde toplanan verilere dayalı olarak yapılan analizler, her bir modelin güçlü ve zayıf yönlerini ortaya koymakta ve bu bilgiler, algoritmaların karmaşık ve dinamik çevrelerde nasıl kararlar aldıklarını daha iyi anlamak için kullanılmıştır.

Modellerin performanslarına genel olarak bakıldığında, TD3 algoritmasının istikrarlı ve yüksek performans sergilediği, DDPG algoritmasının ise daha düşük bir

performans gösterdiği görülmüştür. PPO algoritması başlangıçta iyi bir performans sergilese de zamanla öğrenme ivmesinde dalgalanmalar yaşamıştır. Eğitim sürecinde elde edilen kümülatif ortalama ödül değerleri ve öğrenme eğrileri, TD3'ün zamanla diğer modellere göre daha yüksek ve stabil performans sergilediğini ortaya koymuştur. Bu durum TD3'ün kararlılık ve performans açısından üstün olduğunu göstermektedir.

Eğitim sürecinde elde edilen ortalama bölüm süresi, kazanılan ödül, hız değerleri ve başarılı tur oranları gibi istatistikler, modellerin sürüş stratejilerini ve adaptasyon kabiliyetlerini anlamak için önemli veriler sunmuştur. Örneğin TD3 modeli, en yüksek ortalama ödülle ve hızla performansını diğer modellerden üstün bir şekilde sergilerken; DDPG modeli, daha düşük performans göstermiştir. PPO modeli ise belirli bölümlerde yüksek başarı oranı ve ödüller elde etmesine rağmen genel olarak daha düşük hızda ve daha temkinli bir sürüş politikası izlemiştir.

Test yarışlarında da benzer sonuçlar elde edilmiştir. TD3 modeli, en hızlı tur süresi ve en yüksek ortalama hız gibi metriklerde öne çıkmış, yarış süresince hiç kaza yapmamış ve en stabil performansı sergilemiştir. TD3, agresif ve risk alabilen bir sürüş politikasıyla dikkat çekmiştir. Bu model dinamik sürüş koşullarına hızlı bir şekilde uyum sağlayabilmiş ve yarış boyunca yüksek stabilite ile etkinlik sergileyerek öne çıkmıştır.

DDPG modeli ise özellikle zorlu virajları ve pist dinamiklerini idare etmede güçlük çekmiş daha fazla kaza yapmış ve daha düşük performans sergilemiştir. Bu modelin daha fazla optimizasyona ihtiyaç duyduğu, özellikle sürüş stratejilerinin ve karar verme mekanizmalarının geliştirilmesi gerektiği görülmüştür.

PPO modeli, test yarışında nispeten daha düşük hızda ve daha az riskli bir sürüş politikası benimsemiştir. Bu sürüş politikası, virajlarda ve zorlu pist koşullarında daha az kaza yapmasını sağlamıştır. Ancak genel yarış süreleri ve hız performansı açısından dezavantajlı hale gelmesine neden olmuştur. PPO'nun adaptasyon ve öğrenme sürecinin, belirli pist koşullarına daha iyi uyarlanabilmesi için daha fazla geliştirilmesi gerektiği söylenebilir.

XAI analizleri ile modellerin karar verme süreçleri ve bu süreçlere etki eden faktörler detaylı bir şekilde incelenmiştir. SHAP (Shapley Additive Explanations)

değerleri ile yapılan global analizlerde, TD3 modelinin gaz, frenleme ve direksiyon eylemleri üzerindeki kararlarının, büyük ölçüde araç hızı ve tekerlek hızlarına dayalı olduğu; PPO modelinin ise pist pozisyonu ve sensör mesafelerine daha fazla odaklandığı görülmüştür. Ayrıca modellerin gerçekleştirdiği görevlerin karmaşık yapısı, birden fazla sensör verisinin bu kararları vermede belirleyici olmasında etkili olmuştur. TD3 algoritmasının lokal açıklanabilirlik analizleri de bu durumu desteklemiştir. Ek olarak TD3'ün yüksek hızda ve karmaşık durumlarda sürüş politikalarının başarılı olduğu, uzman insan sürücü gibi performans sergilediği de görülmüştür. Bu farklı stratejiler modellerin sürüş dinamikleri ve karar verme süreçlerindeki çeşitlilikleri yansıtmaktadır.

Sonuç olarak bu bölümde yapılan analizler ve elde edilen bulgular; DRL algoritmalarının eğitim ve test süreçlerindeki performanslarını, adaptasyon kabiliyetlerini ve karar verme mekanizmalarını detaylı bir şekilde ortaya koymuştur. TD3 algoritması genel olarak en iyi performansı sergilerken, PPO ve DDPG algoritmalarının da belirli durumlarda etkili olabileceği ve potansiyelleri gözlemlenmiştir. Bu bulgular otonom araçların gerçek dünya senaryolarına adaptasyonu açısından kritik öneme sahiptir ve gelecekteki çalışmalar için değerli bilgiler sunmaktadır.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında DRL tabanlı otonom sürüş sistemlerinin performansları değerlendirilmiş ve açıklanabilirliği incelenmiştir. Çalışmada, XAI yöntemleri kullanılarak otonom sürüş esnasında alınan kararların daha şeffaf ve anlaşılabilir hale getirilmesi hedeflenmiş, böylece çeşitli düzenlemelere uyum sağlanması ve kullanıcı güveninin arttırılması amaçlanmıştır.

İlk olarak TD3, DDPG ve PPO modelleri TORCS ortamında eğitilip, test edilmiştir. Eğitim ve test süreçlerinde elde edilen bulgular değerlendirildiğinde, uçtan-uca otonom sürüş sistemlerinde TD3 modelinin yüksek performans gösterdiği ve çeşitli sürüş senaryolarında başarılı olduğu görülmüştür. TD3 modeli, PPO ve DDPG'ye göre daha hızlı ve agresif bir sürüş politikası sergilemiş; buna rağmen daha düşük ihlal ve hata oranlarına sahip olmuştur. DDPG modeli zorlu virajlarda ve çeşitli koşullarda zorluk yaşamış ve TD3'ün gerisinde kalmıştır. PPO modeli ise daha düşük hızlarda seyrederek, düşük riskli bir sürüş politikası benimsemiştir.

Eğitim sırasında elde edilen ısı haritaları ve ihlallerin zamana göre yayılımı incelendiğinde, TD3'ün hatalarını eğitimin ilerleyen safhalarında başarıyla düzelttiği görülmüştür. PPO ve DDPG modelleri de zamanla iyileşme göstermiştir. Modellerin test yarışında bir turluk aktivitelerinde, TD3'ün eylemlerinin yumuşak olduğu, DDPG ve PPO'nun ise daha ani ve keskin eylemlerde bulunduğu gözlemlenmiştir. Bu durum test sonuçlarıyla değerlendirildiğinde, TD3'ün daha başarılı olmasını açıklamaktadır.

Ardından tüm modellerin global açıklanabilirlikleri ve en iyi performansı elde eden TD3 modelinin lokal açıklanabilirliğinin analizi yapılmıştır. Modellerin çıktısına etki eden özelliklere bakıldığında: TD3 eylemlerinde araç hızı, teker dönüş hızı ve mesafe sensörü belirleyici olmuştur. DDPG çıktılarında teker dönüş hızı ve mesafe sensörleri etkili olurken, PPO'nun eylemlerinde pist üstü pozisyon, açı ve belirli mesafe sensörleri belirleyici olmuştur. TD3 ve DDPG daha hızlı bir sürüş stratejisi benimserken, PPO daha basit ve dengeli bir şerit ortalama politikası benimsemiştir. TD3 modelinin farklı durumlarda aldığı kararlara bakıldığında ise model çıktısına etki eden özelliklerin değişiklik gösterdiği görülmüştür. TD3, karmaşık durumlarda uzman insan sürücü gibi kararlar vermiş; viraj giriş ve çıkışlarında aracın kontrolünü

sağlamak için uygun eylemleri seçmiştir. Bu durum eğitim aşamasının başlarında yaşanan spin olaylarına karşı başa çıkmayı öğrendiğini göstermektedir.

Bu analizler, modellerin sürüş dinamiklerini, algoritmik karar verme süreçlerini ve adaptasyon kabiliyetlerini iyileştirmek için değerli bilgiler sunmaktadır. SHAP değerleri kullanılarak yapılan analizler, modellerin iç işleyişini açıklamaya yardımcı olmuş ve kararların şeffaflığını artırmıştır. Bu sonuçlar otonom sürüş sistemlerinin güvenilirliğini ve hesap verebilirliğini artırma potansiyeline sahip olduklarını göstermektedir.

Bu tez, DRL ve XAI alanlarına önemli katkılar sağlamaktadır. DRL algoritmalarının otonom sürüş sistemlerine uygulanması ve bu sistemlerin açıklanabilirliği üzerine yapılan çalışmalar, literatürdeki boşlukları doldurmakta ve yeni araştırma yolları açmaktadır. Ayrıca GDPR gibi yasal düzenlemelere uyum sağlanması amacıyla geliştirilen açıklanabilirlik yöntemleri, otonom araç üreticileri ve düzenleyici kurumlar için önemli rehberler sunmaktadır. Bu yöntemlerin otonom sürüş sistemlerinin güvenilirliğini ve kullanıcılar tarafından kabul edilebilirliğini artırması hedeflenmektedir.

Otonom sürüş alanındaki güncel çalışmalar incelendiğinde, çeşitli görevler (şerit değiştirme, rampa birleştirme, rota planlama) için açıklanabilirliklerin sağlanması geliştiriciler ve kullanıcılar açısından önem arz etmektedir. Kullanıcıların bu açıklanabilirlik analizlerini anlayabilmesi için GPT gibi dil modelleri ile entegre edilerek kullanıcı arayüzü sunulabilir. Abu Dabi Otonom Yarış Ligi (A2RL) gibi otonom araç yarışlarında XAI teknikleri kullanarak iyileştirme sağlanabilir ve izleyicilere daha anlamlı veriler sunulabilir. Ayrıca karmaşık sürüş senaryolarından biri olan drift görevlerinde kullanılan otonom sistemlerde de XAI kullanarak açıklama getirilebilir. Simülasyon ortamında elde edilen sonuçlar gerçek dünya koşullarında farklılık gösterebilir. Bu nedenle gelişmiş bir simülasyon ortamı ve gerçek dünyada doğrulanarak bu testlerin yapılması gerekmektedir. Gerçek dünya koşullarında küçük ölçekli araçlar üzerinde yapılması bir diğer araştırma boşluğudur. Yasal ve etik çerçeveler için açıklanabilirlik üzerinde yapılacak düzenlemeler ve bu düzenlemelere entegrasyon için disiplinler arası çalışmalar yürütülebilir.

Bu tez, DRL tabanlı otonom sürüş sistemlerinin açıklanabilirliğini ve performansını değerlendirerek önemli bulgular ortaya koymuştur. Araştırma sonuçları bu teknolojilerin gelecekteki gelişimi ve uygulamaları için önemli bilgiler sunmakta ve literatüre değerli katkılar sağlamaktadır. Otonom sürüş sistemlerinin güvenilirliğini ve şeffaflığını artırarak, bu teknolojilerin daha geniş bir kabul görmesi ve benimsenmesi için temel oluşturulmuştur. Gelecekteki çalışmalar bu alandaki boşlukları doldurarak, otonom araç teknolojilerinin daha da gelişmesine ve yaygınlaşmasına katkıda bulunacaktır. İleride yapılacak çalışmalarda bu bilgiler doğrultusunda daha dengeli, etkin ve güvenilir otonom sürüş sistemleri geliştirilmesi hedeflenmelidir.

7. KAYNAKLAR

A2RL. (2024). Abu Dhabi Autonomous Racing League (A2RL). Retrieved from https://a2rl.io/

Abraham, H., Lee, C., Brady, S., Fitzgerald, C., Mehler, B., Reimer, B. and Coughlin, J. (2017). *Autonomous Vehicles and Alternatives to Driving: Trust, Preferences, and Effects of Age*.

Agarwal, T., Arora, H. and Schneider, J. (2021). Learning Urban Driving Policies using Deep Reinforcement Learning. *2021 Ieee Intelligent Transportation Systems Conference (Itsc)*, 607-614. doi:10.1109/Itsc48978.2021.9564412

Ahmed, M., Abobakr, A., Lim, C. P. and Nahavandi, S. (2022). Policy-Based Reinforcement Learning for Training Autonomous Driving Agents in Urban Areas With Affordance Learning. *Ieee Transactions on Intelligent Transportation Systems, 23*(8), 12562-12571. doi:10.1109/Tits.2021.3115235

Ahmed, M., Lim, C. P. and Nahavandi, S. (2021). A Deep Q-Network Reinforcement Learning-Based Model for Autonomous Driving. *2021 Ieee International Conference on Systems, Man, and Cybernetics (Smc)*, 739-744. doi:10.1109/Smc52423.2021.9658892

Albahri, A. S., Duhaim, A. M., Fadhel, M. A., Alnoor, A., Baqer, N. S., Alzubaidi, L., . . . Deveci, M. (2023). A systematic review of trustworthy and explainable artificial intelligence in healthcare: Assessment of quality, bias risk, and data fusion. *Information Fusion, 96*, 156-191. doi:https://doi.org/10.1016/j.inffus.2023.03.008

Alighanbari, S. and Azad, N. L. (2021). Safe Adaptive Deep Reinforcement Learning for Autonomous Driving in Urban Environments. Additional Filter? How and Where? *Ieee Access, 9*, 141347-141359. doi:10.1109/Access.2021.3119915

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., . . . Bachem, O. (2020). What matters in on-policy reinforcement learning? A large-scale empirical study. arXiv:2006.05990. http://arxiv.org/abs/2006.05990

Anzalone, L., Barra, P., Barra, S., Castiglione, A. and Nappi, M. (2022). An End-to-End Curriculum Learning Approach for Autonomous Driving Scenarios. *Ieee Transactions on Intelligent Transportation Systems, 23*(10), 19817-19826. doi:10.1109/Tits.2022.3160673

Anzalone, L., Barra, S. and Nappi, M. (2021). Reinforced Curriculum Learning for Autonomous Driving in Carla. *2021 Ieee International Conference on Image Processing (Icip)*, 3318-3322. doi:10.1109/Icip42928.2021.9506673

Ashraf, N. M., Mostafa, R. R., Sakr, R. H. and Rashad, M. Z. (2021). Optimizing hyperparameters of deep reinforcement learning for autonomous driving based on whale optimization algorithm. *Plos One, 16*(6). doi:10.1371/journal.pone.0252754

Ashwin, S. H. and Naveen Raj, R. (2023). Deep reinforcement learning for autonomous vehicles: lane keep and overtaking scenarios with collision avoidance. *International Journal of Information Technology, 15*(7), 3541-3553. doi:10.1007/s41870-023-01412-6

Atakishiyev, S., Salameh, M., Yao, H. and Goebel, R. (2023). Towards safe, explainable, and regulated autonomous driving. In *Explainable Artificial Intelligence for Intelligent Transportation Systems* (pp. 32-52): CRC Press.

Atakishiyev, S., Salameh, M., Yao, H. and Goebel, R. (2024). Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. arXiv:2112.11561. https://arxiv.org/abs/2112.11561

Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., . . . Casalino, G. (2021). Drone Deep Reinforcement Learning: A Review. *Electronics, 10*(9), 999. doi:10.3390/electronics10090999

Bacha, A., Reinholtz, C., Wicks, A., Fleming, M., Naik, A., Avitabile, M. and Elder, N. (2004, 3-6 Oct. 2004). *The DARPA Grand Challenge: overview of the Virginia Tech vehicle and experience.* Paper presented at the Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749).

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., . . . Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion, 58*, 82-115. doi:https://doi.org/10.1016/j.inffus.2019.12.012

Baruch, J. (2016). Steer driverless cars towards full automation. *Nature, 536*(7615), 127-127. doi:10.1038/536127a

Behringer, R., Sundareswaran, S., Gregory, B., Elsley, R., Addison, B., Guthmiller, W., . . . Bevly, D. (2004, 14-17 June 2004). *The DARPA grand challenge - development of an autonomous vehicle*. Paper presented at the IEEE Intelligent Vehicles Symposium, 2004.

Berk, R. A. and Bleich, J. (2013). Statistical Procedures for Forecasting Criminal Behavior. *Criminology & Public Policy, 12*(3), 513-544. doi:https://doi.org/10.1111/1745-9133.12047

Bohn, E., Coates, E. M., Moe, S. and Johansen, T. A. (2019). *Deep Reinforcement Learning Attitude Control of Fixed-Wing UAVs Using Proximal*

Policy optimization. Paper presented at the 2019 International Conference on Unmanned Aircraft Systems (ICUAS).

Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Ackel, L. J., Muller, U., . . . Zieba, K. (2018, 21-25 May 2018). *VisualBackProp: Efficient Visualization of CNNs for Autonomous Driving.* Paper presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA).

Bouhamed, O., Ghazzai, H., Besbes, H. and Massoud, Y. (2020). Autonomous UAV Navigation: A DDPG-based Deep Reinforcement Learning Approach. arXiv:2003.10923. https://arxiv.org/abs/2003.10923

Bugeja, K., Spina, S. and Buhagiar, F. (2017). *Telemetry-based optimisation for user training in racing simulators*. Paper presented at the 2017 9th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games).

Cai, P. D., Wang, H. L., Huang, H. Y., Liu, Y. X. and Liu, M. (2021). Vision-Based Autonomous Car Racing Using Deep Imitative Reinforcement Learning. *Ieee Robotics and Automation Letters, 6*(4), 7262-7269. doi:10.1109/Lra.2021.3097345

Carta, S., Podda, A. S., Reforgiato, D., nbsp, Recupero and Stanciu, M. M. (2021). *Explainable AI for Financial Forecasting*. Paper presented at the Machine Learning, Optimization, and Data Science: 7th International Conference, LOD 2021, Grasmere, UK, October 4–8, 2021, Revised Selected Papers, Part II, Grasmere, United Kingdom. https://doi.org/10.1007/978-3-030-95470-3_5

Chang, C. C., Tsai, J. C., Lin, J. H. and Ooi, Y. M. (2021). Autonomous Driving Control Using the DDPG and RDPG Algorithms. *Applied Sciences-Basel, 11*(22). doi:10.3390/app112210659

Chen, J., Wu, T., Shi, M. P. and Jiang, W. (2020). PORF-DDPG: Learning Personalized Autonomous Driving Behavior with Progressively Optimized Reward Function. *Sensors, 20*(19). doi:10.3390/s20195626

Chen, J. Y., Li, S. B. E. and Tomizuka, M. (2022). Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. *Ieee Transactions on Intelligent Transportation Systems, 23*(6), 5068-5078. doi:10.1109/Tits.2020.3046646

Chen, S. Y., Wang, M. L., Song, W. J., Yang, Y., Li, Y. J. and Fu, M. Y. (2020). Stabilization Approaches for Reinforcement Learning-Based End-to-End Autonomous Driving. *Ieee Transactions on Vehicular Technology, 69*(5), 4740-4750. doi:10.1109/Tvt.2020.2979493

Chen, X., Ghadirzadeh, A., Folkesson, J. and Jensfelt, P. (2018). Deep Reinforcement Learning to Acquire Navigation Skills for Wheel-Legged Robots in Complex Environments. *arXiv pre-print server*. doi:1804.10500

Chen, Y. (2018). *Learning-based Lane Following and Changing Behaviors for Autonomous Vehicle.* (Masters Thesis), Carnegie Mellon University, Pittsburgh, PA.

Chen, Y. L., Dong, C. Y., Palanisamy, P., Mudalige, P., Muelling, K. and Dolan, J. M. (2019). Attention-based Hierarchical Deep Reinforcement Learning for Lane Change Behaviors in Autonomous Driving. *2019 Ieee/Cvf Conference on Computer Vision and Pattern Recognition Workshops (Cvprw 2019)*, 1326-1334. doi:10.1109/Cvprw.2019.00172

Chishti, S. O. A., Riaz, S., Zaib, M. B. and Nauman, M. (2018). *Self-driving Cars Using CNN and Q-learning*. Paper presented at the 2018 Ieee 21st International Multi-Topic Conference (Inmic).

Chromik, M., Eiband, M., Buchner, F., Krüger, A. and Butz, A. (2021). *I Think I Get Your Point, AI! The Illusion of Explanatory Depth in Explainable AI*. Paper presented at the Proceedings of the 26th International Conference on Intelligent User Interfaces, College Station, TX, USA. https://doi.org/10.1145/3397481.3450644

Chukamphaeng, N., Pasupa, K., Antenreiter, M. and Auer, P. (2021). *Learning to Drive with Deep Reinforcement Learning*. Paper presented at the 2021 13th International Conference on Knowledge and Smart Technology (Kst-2021). https://ieeexplore.ieee.org/document/9415770/

Cultrera, L., Seidenari, L., Becattini, F., Pala, P. and Bimbo, A. D. (2020, 14-19 June 2020). *Explaining Autonomous Driving by Learning End-to-End Visual Attention.* Paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

DARPA. (2004). Darpa Grand Challenge 2004. Retrieved from http://archive.darpa.mil/grandchallenge04/

Dong, J., Chen, S., Miralinaghi, M., Chen, T., Li, P. and Labi, S. (2023). Why did the AI make that decision? Towards an explainable artificial intelligence (XAI) for autonomous driving systems. *Transportation Research Part C: Emerging Technologies*, *156*, 104358. doi:https://doi.org/10.1016/j.trc.2023.104358

Doshi-Velez, F. and Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. (arXiv:1702.08608). https://ui.adsabs.harvard.edu/abs/2017arXiv170208608D doi:10.48550/arXiv.1702.08608

European, C., Directorate-General for, R. and Innovation. (2020). *Ethics of connected and automated vehicles – Recommendations on road safety, privacy, fairness, explainability and responsibility*: Publications Office.

Fan, T., Long, P., Liu, W. and Pan, J. (2018). Fully Distributed Multi-Robot Collision Avoidance via Deep Reinforcement Learning for Safe and Efficient

Navigation in Complex Scenarios. arXiv:1808.03841. https://arxiv.org/abs/1808.03841

Fayjie, A. R., Hossain, S., Oualid, D. and Lee, D. J. (2018). Driverless Car: Autonomous Driving Using Deep Reinforcement Learning In Urban Environment. *2018 15th International Conference on Ubiquitous Robots (Ur)*, 896-901.

Fortunato, M., Mohammad, Piot, B., Menick, J., Osband, I., Graves, A., . . . Legg, S. (2019). Noisy Networks for Exploration. arXiv:1706.10295. https://arxiv.org/abs/1706.10295

Fujimoto, S., Herke and Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. *arXiv pre-print server*. doi:1802.09477

GDP, R. (2018). *General data protection regulation (GDPR)*. Intersoft Consulting

Goodman, B. and Flaxman, S. (2017). European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation". *AI Magazine, 38*(3), 50-57. doi:10.1609/aimag.v38i3.2741

Górski, Ł. and Ramakrishna, S. (2021). *Explainable artificial intelligence, lawyer's perspective*. Paper presented at the Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, São Paulo, Brazil. https://doi.org/10.1145/3462757.3466145

Greenblatt, N. A. (2016). Self-driving cars and the law. *IEEE Spectrum, 53*(2), 46-51. doi:10.1109/MSPEC.2016.7419800

Greydanus, S., Koul, A., Dodge, J. and Fern, A. (2017). Visualizing and understanding Atari agents. arXiv:1711.00138. http://arxiv.org/abs/1711.00138

Gu, W. Y., Xu, X. and Yang, J. (2017). Path Following with Supervised Deep Reinforcement Learning. *Proceedings 2017 4th Iapr Asian Conference on Pattern Recognition (Acpr)*, 448-452. doi:10.1109/Acpr.2017.30

Guckiran, K. and Bolat, B. (2019). *Autonomous Car Racing in Simulation Environment Using Deep Reinforcement Learning*. Paper presented at the 2019 Innovations in Intelligent Systems and Applications Conference (Asyu).

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. and Pedreschi, D. (2018). A Survey of Methods for Explaining Black Box Models. *51*(5 %J ACM Comput. Surv.), Article 93. doi:10.1145/3236009

He, L., Aouf, N. and Song, B. (2021). Explainable Deep Reinforcement Learning for UAV autonomous path planning. *Aerospace Science and Technology, 118*, 107052. doi:https://doi.org/10.1016/j.ast.2021.107052

Hessel, M., Modayil, J., Hado, Schaul, T., Ostrovski, G., Dabney, W., . . . Silver, D. (2017). Rainbow: Combining Improvements in Deep Reinforcement Learning. arXiv:1710.02298. https://arxiv.org/abs/1710.02298

Heuillet, A., Couthouis, F. and Díaz-Rodríguez, N. (2020). Explainability in deep reinforcement learning. arXiv:2008.06693. http://arxiv.org/abs/2008.06693

Hilleli, B. and El-Yaniv, R. (2018). Toward Deep Reinforcement Learning without a Simulator: An Autonomous Steering Example. *Thirty-Second Aaai Conference on Artificial Intelligence / Thirtieth Innovative Applications of Artificial Intelligence Conference / Eighth Aaai Symposium on Educational Advances in Artificial Intelligence*, 1471-1478.

Hoffman, R. R. and Klein, G. (2017). Explaining Explanation, Part 1: Theoretical Foundations. *IEEE Intelligent Systems, 32*(3), 68-73. doi:10.1109/MIS.2017.54

Honda. (2021). Honda Sustainability Report 2021. In: Social Contribution Activities Minato, Tokyo.

Hossain, J. (2023). Autonomous driving with deep reinforcement learning in CARLA simulation. arXiv:2306.11217. http://arxiv.org/abs/2306.11217

Huang, C. X., Zhang, R. H., Ouyang, M. Z., Wei, P. X., Lin, J. F., Su, J. and Lin, L. (2021). Deductive Reinforcement Learning for Visual Autonomous Urban Driving Navigation. *Ieee Transactions on Neural Networks and Learning Systems, 32*(12), 5379-5391. doi:10.1109/Tnnls.2021.3109284

Huang, Z., Wu, J. and Lv, C. (2023). Efficient Deep Reinforcement Learning With Imitative Expert Priors for Autonomous Driving. *Ieee Transactions on Neural Networks and Learning Systems, 34*(10), 7391-7403. doi:10.1109/TNNLS.2022.3142822

Huang, Z. Q., Zhang, J., Tian, R. and Zhang, Y. X. (2019). End-to-End Autonomous Driving Decision Based on Deep Reinforcement Learning. *Conference Proceedings of 2019 5th International Conference on Control, Automation and Robotics (Iccar)*, 658-662.

Hussonnois, M. and Jun, J. Y. (2022). *End-to-end autonomous driving using the Ape-X algorithm in Carla simulation environment*. Paper presented at the 2022 Thirteenth International Conference on Ubiquitous and Future Networks (Icufn). https://ieeexplore.ieee.org/document/9829674/

Iyer, R., Li, Y., Li, H., Lewis, M., Sundar, R. and Sycara, K. (2018). *Transparency and Explanation in Deep Reinforcement Learning Neural Networks*. Paper presented at the Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, New Orleans, LA, USA. https://doi.org/10.1145/3278721.3278776

Jacovi, A. and Goldberg, Y. (2020). Towards Faithfully Interpretable NLP Systems: How should we define and evaluate faithfulness? arXiv:2004.03685. http://arxiv.org/abs/2004.03685

Jaritz, M., Raoul, Toromanoff, M., Perot, E. and Nashashibi, F. (2018). End-to-End Race Driving with Deep Reinforcement Learning. arXiv:1807.02371. https://arxiv.org/abs/1807.02371

Jin, Y. L., Liu, Q. H., Shen, L. Q. and Zhu, L. J. (2021). Deep Deterministic Policy Gradient Algorithm Based on Convolutional Block Attention for Autonomous Driving. *Symmetry-Basel, 13*(6). doi:10.3390/sym13061061

Kaelbling, L. P., Littman, M. L. and Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research, 4*, 237-285. doi:10.1613/jair.301

Kaushik, M., Prasad, V., Krishna, K. M. and Ravindran, B. (2018, 26-30 June 2018). *Overtaking Maneuvers in Simulated Highway Driving using Deep Reinforcement Learning.* Paper presented at the 2018 IEEE Intelligent Vehicles Symposium (IV).

Kim, J. and Canny, J. (2017, 22-29 Oct. 2017). *Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention.* Paper presented at the 2017 IEEE International Conference on Computer Vision (ICCV).

Kim, J., Rohrbach, A., Darrell, T., Canny, J. and Akata, Z. (2018, 2018//). *Textual Explanations for Self-Driving Vehicles.* Paper presented at the Computer Vision – ECCV 2018, Cham.

Kishikawa, D. and Arai, S. (2021). Estimation of personal driving style via deep inverse reinforcement learning. *Artificial Life and Robotics, 26*(3), 338-346. doi:10.1007/s10015-021-00682-2

Koch, W., Mancuso, R., West, R. and Bestavros, A. (2019). Reinforcement Learning for UAV Attitude Control. *ACM Trans. Cyber-Phys. Syst., 3*(Association for Computing Machinery), Article 22. doi:10.1145/3301273

Kök, İ., Okay, F. Y., Ö, M. and Özdemir, S. (2023). Explainable Artificial Intelligence (XAI) for Internet of Things: A Survey. *Ieee Internet of Things Journal, 10*(16), 14764-14779. doi:10.1109/JIOT.2023.3287678

Kurshan, E., Chen, J., Storchan, V. and Shen, H. (2022). *On the current and emerging challenges of developing fair and ethical AI solutions in financial services*. Paper presented at the Proceedings of the Second ACM International Conference on AI in Finance, Virtual Event. https://doi.org/10.1145/3490354.3494408

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436-444. doi:10.1038/nature14539

Lee, J. D. and See, K. A. (2004). Trust in Automation: Designing for Appropriate Reliance. *46*(1), 50-80. doi:10.1518/hfes.46.1.50_30392

Lee, K., Vlahov, B., Gibson, J., Rehg, J. M. and Theodorou, E. A. (2021). *Approximate Inverse Reinforcement Learning from Vision-based Imitation Learning*. Paper presented at the 2021 Ieee International Conference on Robotics and Automation (Icra 2021).

Li, D., Zhao, D. B., Zhang, Q. C. and Chen, Y. R. (2019). Reinforcement Learning and Deep Learning Based Lateral Control for Autonomous Driving. *Ieee Computational Intelligence Magazine, 14*(2), 83-98. doi:10.1109/Mci.2019.2901089

Li, D. F., Meng, L. C., Li, J. J., Lu, K. and Yang, Y. (2022). Domain adaptive state representation alignment for reinforcement learning. *Information Sciences, 609*, 1353-1368. doi:10.1016/j.ins.2022.07.156

Liang, X. D., Wang, T. R., Yang, L. N. and Xing, E. R. (2018). CIRL: Controllable Imitative Reinforcement Learning for Vision-Based Self-driving. *Computer Vision - Eccv 2018, Pt Vii, 11211*, 604-620. doi:10.1007/978-3-030-01234-2_36

Lin, J. Q., Zhang, P., Li, C. G., Zhou, Y. P., Wang, H. J. and Zou, X. J. (2022). APF-DPPO: An Automatic Driving Policy Learning Method Based on the Artificial Potential Field Method to Optimize the Reward Function. *Machines, 10*(7). doi:10.3390/machines10070533

Liu, H. C., Huang, Z. Y., Wu, J. D. and Lv, C. (2022). *Improved Deep Reinforcement Learning with Expert Demonstrations for Urban Autonomous Driving*. Paper presented at the 2022 Ieee Intelligent Vehicles Symposium (Iv). https://ieeexplore.ieee.org/document/9827073/

Liu, K., Wan, Q. and Li, Y. J. (2018). *A Deep Reinforcement Learning Algorithm with Expert Demonstrations and Supervised Loss and its application in Autonomous Driving*. Paper presented at the 2018 37th Chinese Control Conference (Ccc).

Loiacono, D., Cardamone, L. and Lanzi, P. L. (2013). Simulated Car Racing Championship: Competition Software Manual. arXiv:1304.1672. https://arxiv.org/abs/1304.1672

Lundberg, S. M. and Lee, S.-I. (2017). *A unified approach to interpreting model predictions*. Paper presented at the Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA.

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., . . . Lee, S.-I. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering, 2*(10), 749-760. doi:10.1038/s41551-018-0304-0

Luo, M., Tong, Y. and Liu, J. C. (2018). *Orthogonal Policy Gradient and Autonomous Driving Application*. Paper presented at the Proceedings of 2018 Ieee 9th International Conference on Software Engineering and Service Science (Icsess).

Mandeep, Agarwal, A., Bhatia, A., Malhi, A., Kaler, P. and Pannu, H. S. (2022, 1-3 July 2022). *Machine Learning Based Explainable Financial Forecasting.* Paper presented at the 2022 4th International Conference on Computer Communication and the Internet (ICCCI).

Mankodiya, H., Obaidat, M. S., Gupta, R. and Tanwar, S. (2021, 15-17 Oct. 2021). *XAI-AV: Explainable Artificial Intelligence for Trust Management in Autonomous Vehicles.* Paper presented at the 2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI).

Marc, Dabney, W. and Munos, R. e. (2017). A Distributional Perspective on Reinforcement Learning. arXiv:1707.06887. https://arxiv.org/abs/1707.06887

Mardaoui, D. and Garreau, D. (2020). An analysis of LIME for text data. arXiv:2010.12487. http://arxiv.org/abs/2010.12487

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. arXiv:1312.5602. https://arxiv.org/abs/1312.5602

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., . . . Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529-533. doi:10.1038/nature14236

Montavon, G., Samek, W. and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing, 73*, 1-15. doi:https://doi.org/10.1016/j.dsp.2017.10.011

Mori, K., Fukui, H., Murase, T., Hirakawa, T., Yamashita, T. and Fujiyoshi, H. (2019, 9-12 June 2019). *Visual Explanation by Attention Branch Network for End-to-end Learning-based Self-driving.* Paper presented at the 2019 IEEE Intelligent Vehicles Symposium (IV).

Mulgan, R. (2000). 'Accountability': An Ever-Expanding Concept? *Public Administration, 78*(3), 555-573. doi:https://doi.org/10.1111/1467-9299.00218

Nahata, R., Omeiza, D., Howard, R. and Kunze, L. (2021, 19-22 Sept. 2021). *Assessing and Explaining Collision Risk in Dynamic Environments for Autonomous Driving Safety.* Paper presented at the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC).

Nazat, S., Li, L. and Abdallah, M. (2024). XAI-ADS: An Explainable Artificial Intelligence Framework for Enhancing Anomaly Detection in Autonomous Driving Systems. *Ieee Access, 12*, 48583-48607. doi:10.1109/ACCESS.2024.3383431

Niu, J. Y., Hu, Y., Jin, B. B., Han, Y. H. and Li, X. W. (2020). Two-Stage Safe Reinforcement Learning for High-Speed Autonomous Racing. *2020 Ieee International Conference on Systems, Man, and Cybernetics (Smc)*, 3934-3941.

NTSB. (2018). Collision Between a Sport Utility Vehicle Operating With Partial Driving Automation and a Crash Attenuator. Retrieved from https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR2001.pdf

Nunes, A. and Axhausen, K. W. (2021). Road safety, health inequity and the imminence of autonomous vehicles. *Nature Machine Intelligence, 3*(8), 654-655. doi:10.1038/s42256-021-00382-3

Omeiza, D., Webb, H., Jirotka, M. and Kunze, L. (2022). Explanations in Autonomous Driving: A Survey. *Ieee Transactions on Intelligent Transportation Systems, 23*(8), 10142-10162. doi:10.1109/TITS.2021.3122865

OpenAI, Andrychowicz, M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., . . . Zaremba, W. (2019). Learning Dexterous In-Hand Manipulation. arXiv:1808.00177. https://arxiv.org/abs/1808.00177

Ozguner, U., Redmill, K. A. and Broggi, A. (2004, 14-17 June 2004). *Team TerraMax and the DARPA grand challenge: a general overview*. Paper presented at the IEEE Intelligent Vehicles Symposium, 2004.

Pan, X., You, Y., Wang, Z. and Lu, C. (2017). Virtual to Real Reinforcement Learning for Autonomous Driving. arXiv:1704.03952. https://arxiv.org/abs/1704.03952

Peng, B. Y., Sun, Q., Li, S. E., Kum, D., Yin, Y. M., Wei, J. Q. and Gu, T. Y. (2021). End-to-End Autonomous Driving Through Dueling Double Deep Q-Network. *Automotive Innovation, 4*(3), 328-337. doi:10.1007/s42154-021-00151-3

Peng, M. X., Gong, Z. H., Sun, C., Chen, L. and Cao, D. P. (2020). *Imitative Reinforcement Learning Fusing Vision and Pure Pursuit for Self-driving*. Paper presented at the 2020 Ieee International Conference on Robotics and Automation (Icra).

Perez-Gill, O., Barea, R., Lopez-Guillen, E., Bergasa, L. M., Gomez-Huelamo, C., Gutierrez, R. and Diaz, A. (2021). *Deep Reinforcement Learning based control algorithms: Training and validation using the ROS Framework in CARLA Simulator for Self-Driving applications*. Paper presented at the 2021 32nd Ieee Intelligent Vehicles Symposium (Iv). https://ieeexplore.ieee.org/document/9575616/

Plaat, A. (2022). Deep reinforcement learning, a textbook. arXiv:2201.02135. http://arxiv.org/abs/2201.02135

Popov, I., Heess, N., Lillicrap, T., Hafner, R., Barth-Maron, G., Vecerik, M., . . . Riedmiller, M. (2017). Data-efficient Deep Reinforcement Learning for Dexterous Manipulation. arXiv:1704.03073. https://arxiv.org/abs/1704.03073

Quek, Y. T., Koh, L., Koh, N. T., Tso, W. A. and Woo, W. L. (2021). Deep Q-network implementation for simulated autonomous vehicle control. *Iet Intelligent Transport Systems, 15*(7), 875-885. doi:10.1049/itr2.12067

Rahimpour, A., Martin, S., Tawari, A. and Qi, H. (2019, 9-12 June 2019). *Context Aware Road-user Importance Estimation (iCARE).* Paper presented at the 2019 IEEE Intelligent Vehicles Symposium (IV).

Remman, S. B., Strümke, I. and Lekkas, A. M. (2021). Causal versus marginal Shapley values for robotic lever manipulation controlled using deep reinforcement learning. arXiv:2111.02936. http://arxiv.org/abs/2111.02936

Riboni, A., Candelieri, A. and Borrotti, M. (2022). Deep Autonomous Agents Comparison for Self-driving Cars. *Machine Learning, Optimization, and Data Science (Lod 2021), Pt I, 13163*, 201-213. doi:10.1007/978-3-030-95467-3_16

SAE. (2021). Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. In: SAE International On-Road Automated Driving Committee.

Sallab, A. E., Abdou, M., Perot, E. and Yogamani, S. (2017). Deep Reinforcement Learning framework for Autonomous Driving. *Electronic Imaging, 29*(19), 70-76. doi:10.2352/issn.2470-1173.2017.19.avm-023

Savari, M. and Choe, Y. (2022). Utilizing Human Feedback in Autonomous Driving: Discrete vs. Continuous. *Machines, 10*(8). doi:10.3390/machines10080609

Schaul, T., Quan, J., Antonoglou, I. and Silver, D. (2016). Prioritized Experience Replay. arXiv:1511.05952. https://arxiv.org/abs/1511.05952

Schneider, T., Hois, J., Rosenstein, A., Ghellal, S., Theofanou-Fülbier, D. and Gerlicher, A. R. S. (2021). *ExplAIn Yourself! Transparency for Positive UX in Autonomous Driving*. Paper presented at the Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan. https://doi.org/10.1145/3411764.3446647

Schulman, J., Levine, S., Moritz, P., Michael and Abbeel, P. (2017a). Trust Region Policy Optimization. *arXiv pre-print server*. doi:1502.05477

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017b). Proximal Policy Optimization Algorithms. *arXiv pre-print server*. doi:1707.06347

Sharifzadeh, S., Chiotellis, I., Triebel, R. and Cremers, D. (2017). Learning to Drive using Inverse Reinforcement Learning and Deep Q-Networks. arXiv:1612.03653. https://arxiv.org/abs/1612.03653

Shen, Y., Jiang, S., Chen, Y. and Campbell, K. D. (2022). To explain or not to explain: A study on the necessity of explanations for autonomous vehicles. arXiv:2006.11684. http://arxiv.org/abs/2006.11684

Siboo, S., Bhattacharyya, A., Raj, R. N. and Ashwin, S. H. (2023). An Empirical Study of DDPG and PPO-Based Reinforcement Learning Algorithms for Autonomous Driving. *Ieee Access, 11*, 125094-125108. doi:10.1109/ACCESS.2023.3330665

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., . . . Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484-489. doi:10.1038/nature16961

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D. and Riedmiller, M. (2014). *Deterministic policy gradient algorithms*. Paper presented at the Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, Beijing, China.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., . . . Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354-359. doi:10.1038/nature24270

Smith, C. (1978). *Tune To Win*. United States: Aero Publishers.

Stang, M., Grimm, D., Gaiser, M. and Sax, E. (2020). *Evaluation of Deep Reinforcement Learning Algorithms for Autonomous Driving*. Paper presented at the 2020 Ieee Intelligent Vehicles Symposium (Iv).

Tai, L., Paolo, G. and Liu, M. (2017). Virtual-to-real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation. arXiv:1703.00420. https://arxiv.org/abs/1703.00420

Tammewar, A., Chaudhari, N., Saini, B., Venkatesh, D., Dharahas, G., Vora, D., . . . Alfarhood, S. (2023). Improving the Performance of Autonomous Driving through Deep Reinforcement Learning. *15*(18), 13799.

Tian, Y. T., Cao, X. H., Huang, K., Fei, C., Zheng, Z. and Ji, X. W. (2022). Learning to Drive Like Human Beings: A Method Based on Deep Reinforcement Learning. *Ieee Transactions on Intelligent Transportation Systems*, *23*(7), 6357-6367. doi:10.1109/Tits.2021.3055899

Timothy, Jonathan, Pritzel, A., Heess, N., Erez, T., Tassa, Y., . . . Wierstra, D. (2019). Continuous control with deep reinforcement learning. arXiv:1509.02971. https://arxiv.org/abs/1509.02971

Tomsett, R., Preece, A., Braines, D., Cerutti, F., Chakraborty, S., Srivastava, M., . . . Kaplan, L. (2020). Rapid Trust Calibration through Interpretable and Uncertainty-Aware AI. *Patterns, 1*(4), 100049. doi:https://doi.org/10.1016/j.patter.2020.100049

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the Theory of the Brownian Motion. *Physical Review, 36*(5), 823-841. doi:10.1103/PhysRev.36.823

van Hasselt, H., Guez, A. and Silver, D. (2016). *Deep Reinforcement Learning with Double Q-Learning*. Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence. https://ojs.aaai.org/index.php/AAAI/article/view/10295

Voigt, P. and von dem Bussche, A. (2017). *The EU general data protection regulation (GDPR)* (1 ed.). Cham, Switzerland: Springer International Publishing.

Wang, C., Wu, L., Yan, C., Wang, Z., Long, H. and Yu, C. (2020). Coactive design of explainable agent-based task planning and deep reinforcement learning for human-UAVs teamwork. *Chinese Journal of Aeronautics, 33*(11), 2930-2945. doi:https://doi.org/10.1016/j.cja.2020.05.001

Wang, T. H., Luo, Y. G., Liu, J. X. and Li, K. Q. (2021). Multi-Objective End-to-End Self-Driving Based on Pareto-Optimal Actor-Critic Approach. *2021 Ieee Intelligent Transportation Systems Conference (Itsc)*, 473-478. doi:10.1109/Itsc48978.2021.9564464

Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., . . . Miao, Q. (2024). Deep Reinforcement Learning: A Survey. *Ieee Transactions on Neural Networks and Learning Systems, 35*(4), 5064-5078. doi:10.1109/TNNLS.2022.3207346

Wang, Z., Schaul, T., Hessel, M., Hado, Lanctot, M. and Nando. (2016). Dueling Network Architectures for Deep Reinforcement Learning. arXiv:1511.06581. https://arxiv.org/abs/1511.06581

Wang, Z. L., Wang, H., Cui, X. and Zheng, C. C. (2021). *A Multi-sensing Input and Multi-constraint Reward Mechanism Based Deep Reinforcement Learning Method for Self-driving Policy Learning*. Paper presented at the Intelligent Robotics and Applications, Icira 2021, Pt Iv. https://link.springer.com/chapter/10.1007/978-3-030-89092-6_63

Wu, H., Song, S., You, K. and Wu, C. (2019). Depth Control of Model-Free AUVs via Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 49*(12), 2499-2510. doi:10.1109/TSMC.2017.2785794

Wu, Y. Q., Liao, S. Q., Liu, X., Li, Z. H. and Lu, R. Q. (2021). Deep Reinforcement Learning on Autonomous Driving Policy With Auxiliary Critic Network. *Ieee Transactions on Neural Networks and Learning Systems*. doi:10.1109/Tnnls.2021.3116063

Wymann, B. TORCS. Retrieved from https://torcs.sourceforge.net/

Xia, W., Li, H. Y. and Li, B. P. (2016). *A Control Strategy of Autonomous Vehicles based on Deep Reinforcement Learning*. Paper presented at the Proceedings of 2016 9th International Symposium on Computational Intelligence and Design (Iscid), Vol 2.

Xu, J., Du, T., Foshey, M., Li, B., Zhu, B., Schulz, A. and Matusik, W. (2019). Learning to fly: computational controller design for hybrid UAVs with reinforcement learning. *ACM Trans. Graph., 38*(Association for Computing Machinery), Article 42. doi:10.1145/3306346.3322940

Xu, J. X. and Yuan, J. (2018). HeuRL: A Heuristically Initialized Reinforcement Learning Method for Autonomous Driving Control Task. *2018 International Conference on Control and Robots (Iccr)*, 57-62.

Xu, Y., Yang, X., Gong, L., Lin, H. C., Wu, T. Y., Li, Y. and Vasconcelos, N. (2020, 13-19 June 2020). *Explainable Object-Induced Action Decision for Autonomous Vehicles.* Paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Yang, F., Wang, P. and Wang, X. H. (2018). *Continuous Control in Car Simulator with Deep Reinforcement Learning*. Paper presented at the Proceedings of 2018 the 2nd International Conference on Computer Science and Artificial Intelligence (Csai 2018) / 2018 the 10th International Conference on Information and Multimedia Technology (Icimt 2018).

Yang, R. Y., Tang, H. Y., Jin, B. H. and Liu, K. C. (2021). *Columba: A New Approach to Train an Agent for Autonomous Driving*. Paper presented at the 2021 International Joint Conference on Neural Networks (Ijcnn). https://ieeexplore.ieee.org/document/9533604/

Ye, C. K., Ma, H. M., Zhang, X. Q., Zhang, K. and You, S. D. (2017). *Survival-Oriented Reinforcement Learning Model: An Effcient and Robust Deep Reinforcement Learning Algorithm for Autonomous Driving Problem*. Paper presented at the Image and Graphics (Icig 2017), Pt Ii.

Yu, R., Shi, Z., Huang, C., Li, T. and Ma, Q. (2017, 26-28 July 2017). *Deep reinforcement learning based optimal trajectory tracking control of autonomous underwater vehicle.* Paper presented at the 2017 36th Chinese Control Conference (CCC).

Yuan, Y. L., Tasik, R., Adhatarao, S. S., Yuan, Y. C., Liu, Z. L. and Fu, X. M. (2020). RACE: Reinforced Cooperative Autonomous Vehicle Collision Avoidance. *Ieee Transactions on Vehicular Technology, 69*(9), 9279-9291. doi:10.1109/Tvt.2020.2974133

Zhang, Y., Tiňo, P., Leonardis, A. and Tang, K. (2020). A Survey on Neural Network Interpretability. arXiv:2012.14261. http://arxiv.org/abs/2012.14261

Zhao, J., Zhao, W., Deng, B., Wang, Z., Zhang, F., Zheng, W., . . . Burke, A. F. (2024). Autonomous driving system: A comprehensive survey. *Expert Systems with Applications, 242*, 122836. doi:https://doi.org/10.1016/j.eswa.2023.122836

Zhu, G. S., Pei, C. M., Ding, J. and Shi, J. F. (2020). *Deep Deterministic Policy Gradient Algorithm based Lateral and Longitudinal Control for Autonomous Driving*. Paper presented at the 2020 5th International Conference on Mechanical, Control and Computer Engineering (Icmcce 2020). https://ieeexplore.ieee.org/document/9421545/

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L. and Farhadi, A. (2016). Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. arXiv:1609.05143. https://arxiv.org/abs/1609.05143

Zou, Q. J., Xiong, K., Fang, Q. and Jiang, B. H. (2021). Deep imitation reinforcement learning for self-driving by vision. *Caai Transactions on Intelligence Technology, 6*(4), 493-503. doi:10.1049/cit2.12025

Zou, Q. J., Xiong, K. and Hou, Y. L. (2020). *An end-to-end learning of driving strategies based on DDPG and imitation learning*. Paper presented at the Proceedings of the 32nd 2020 Chinese Control and Decision Conference (Ccdc 2020).

Zuo, S. X., Wang, Z. Y., Zhu, X. R. and Ou, Y. S. (2017). *Continuous Reinforcement Learning From Human Demonstrations With Integrated Experience Replay for Autonomous Driving*. Paper presented at the 2017 Ieee International Conference on Robotics and Biomimetics (Ieee Robio 2017).

EKLER

8. EKLER

EK A

Tablo 8.1: Eğitim sürecinde gerçekleşen tüm hata ve ihlallerin her bir etaptaki (bir etap 1000 bölüm içerir) sayıları.

Tüm hatalar	Etap No. (1 etap = 1000 bölüm)	TD3	DDPG	PPO
Kaza	1	962	660	732
	2	114	353	239
	3	113	259	18
	4	122	223	27
Pist limiti ihlali	1	2961	3801	3078
	2	1752	5144	3477
	3	884	4049	575
	4	1026	3240	1485
İlerleme yok / çok yavaşlama	1	60	554	450
	2	229	299	1316
	3	137	784	821
	4	190	973	172
Spin veya ters yönde ilerleme	1	7	144	17
	2	242	194	3
	3	137	214	12
	4	169	190	21

EK B

Tablo 8.2: Eğitim süresince gerçekleşen sonlandırma kriterlerinin etap başına sayıları.

Tüm hatalar	Etap No. (1 etap = 1000 bölüm)	TD3	DDPG	PPO
Kaza	1	948	643	722
	2	103	339	239
	3	104	245	18
	4	112	213	26
Pist limiti ihlali	1	40	118	94
	2	87	245	332
	3	22	144	12
	4	26	158	45
İlerleme yok / çok yavaşlama	1	3	84	68
	2	4	19	173
	3	8	88	126
	4	8	113	7
Spin veya ters yönde ilerleme	1	7	144	17
	2	242	194	3
	3	137	214	12
	4	169	190	21
Başarıyla tamamlanan tur	1	2	11	99
	2	564	203	253
	3	729	309	832
	4	685	326	901