

# Satranç Bulmaca Zorluk Düzeyi Tahmini

Seyyid Osman Sevgili  
Bilgisayar Mühendisliği  
İstanbul Teknik Üniversitesi  
İstanbul, Türkiye  
sevgili17@itu.edu.tr

**Özet**—Bu çalışmada, IEEE Büyük Veri Kupası kapsamında sağlanan veri kümesi kullanılarak satranç bulmacalarının zorluk seviyelerini tahmin etmek amaçlanmıştır. Veri kümesi, Forsyth-Edwards Notasyonu (FEN) ile ifade edilen satranç tahtası pozisyonları, Portable Game Notation (PGN) formatında bulmaca çözümleri ve Glicko-2 algoritması ile belirlenen zorluk seviyelerini içermektedir. Çalışmada, bulmaca zorluklarını tahmin etmek için iki farklı yöntem önerilmiş ve değerlendirilmiştir: LightGBM (Light Gradient Boosting Machine) ile lineer özelliklerin yanı sıra geçerli hamle ve tehdit çizgelerinden elde edilen özellikleri kullanan yaklaşım ve CNN-GRU tabanlı özelleştirilmiş sinir ağı modeli.

Özelleştirilmiş sinir ağı modeli, test veri kümesi üzerinde 142.150 ortalama karesel hata (MSE) değeri ile diğer yöntemlerden daha iyi performans göstermiştir. Lineer yaklaşım 176.159 MSE, çizge tabanlı yaklaşım ise 163.668 MSE değeri elde etmiştir. Önerilen CNN-GRU modeli, satranç bulmacalarının karmaşıklığını daha iyi kavrayarak zorluk seviyelerini doğru bir şekilde tahmin edebilmiştir. Ayrıca, veri kümesi üzerinde gerçekleştirilen keşifsel veri analizi ile gürültülü veriler temizlenmiş ve modelin eğitim güvenilirliği artırılmıştır.

**Anahtar Sözcükler**—Çizge, CNN-GRU, FEN, PGN, Yapay Sinir Ağı, LightGBM

## I. GİRİŞ

Satranç, tarihi yaklaşık 1500 yıl öncesine dayanan, 8x8 karelik bir tahta üzerinde piyon, at, fil, kale, vezir ve şah gibi çeşitli yeteneklere sahip taşlarla oynanan bir strateji oyunudur [1]. Bir satranç oyunundaki olası hamle kombinasyonlarının alt sınırı, Shannon sayısı olarak bilinen ve  $10^{120}$  değerine eşit olan astronomik bir büyüklüktedir [2]. Bu durum, satrancın derin ve karmaşık stratejik yapısını ortaya koymaktadır.

Tarih boyunca satranç oyuncuları, çeşitli stratejiler geliştirerek rakiplerine üstünlük sağlamayı hedeflemiştir. Günümüzde ise bilgisayar teknolojilerinin gelişimi, oyuncuların strateji geliştirme süreçlerine yapay zeka temelli algoritmaların entegre edilmesini mümkün kılmıştır. Satranç oyuncularının yeteneklerini geliştirmelerinin en etkili yollarından biri, farklı oyun pozisyonlarındaki kazanç sağlayan hamleleri bulmaya yönelik pratikler yapmaktır. Bu doğrultuda, günümüzde teknoloji sayesinde oyun sonu, oyun ortası ve açılış pozisyonlarını içeren bulmacalar oluşturulmakta ve oyuncuların bu bulmacaları çözerek kendilerini geliştirmeleri hedeflenmektedir. Ayrıca, satranç bulmacaları yalnızca eğlence amaçlı da çözülebilmekte ve oyunculara keyifli bir deneyim sunmaktadır.

Satranç bulmacaları oluşturmak kolay olsa da, oluşturulan bir satranç bulmacasının zorluk düzeyini belirlemek oldukça

karmaşık bir problemidir. Zorluk düzeyinin doğru bir şekilde tahmin edilmesi, hem oyuncu deneyimi hem de oyuncuların gelişimi açısından kritik öneme sahiptir. Günümüzde satranç bulmacalarının zorluk düzeyini belirlemek için yaygın olarak kullanılan yöntemlerden biri Glicko-2 algoritmasıdır [3]. Bu algoritma, oyuncuların ve bulmacaların performanslarını değerlendirmek için etkili bir çözüm sunmaktadır. Ancak, Glicko-2 algoritması kullanıcı girdisine yüksek derecede bağımlıdır ve bu durum, özellikle yanlış veya eksik verilerle karşılaşıldığında tahminlerin doğruluğunu olumsuz etkileyebilmektedir. Bu durum, daha bağımsız ve doğru bir tahmin sistemi geliştirme ihtiyacını doğurmuştur.

Bu çalışmada, satranç bulmacalarının zorluk düzeyini tahmin etmeye yönelik iki farklı çözüm yöntemi ele alınmıştır. İlk yöntemde, çeşitli özellik mühendisliği teknikleri kullanılarak satranç tahtasındaki pozisyonları en iyi temsil eden özellikler ortaya çıkarılmış ve bu özellikler, LightGBM algoritması kullanılarak değerlendirilmiştir. İkinci yaklaşım ise derin öğrenme yöntemlerine dayalı olarak, bir CNN (Convolutional Neural Network) ve GRU (Gated Recurrent Unit) birleşiminden oluşan hibrit bir yapay sinir ağı tasarımı geliştirilmiştir. Önerilen bu yaklaşımlarla, satranç bulmacalarının zorluk düzeyini belirlemede kullanıcı bağımlılığı ortadan kaldırılarak, zorluk düzeyinin doğru bir şekilde tahmin edilmesi ve böylece kullanıcı deneyimi ile oyuncu gelişiminin üst düzeye çıkarılması hedeflenmiştir.

Geliştirilen sistem, kullanıcı girdisinden bağımsız olarak, verilen satranç bulmacasını derecelendirme yeteneğine sahiptir. Önerilen sistem, satranç eğitimi platformlarında ve turnuva hazırlık süreçlerinde oyuncuların gelişimini desteklemek için uygulanabilir. Ayrıca, sistemin performansı, literatürde önerilen diğer sistemlerle benzer bir düzeyde elde edilmiştir.

Makale aşağıdaki şekilde yapılandırılmıştır. II. bölümde, satranç bulmacalarının zorluk derecelerini tahmin etme konusunda yapılan mevcut çalışmalar ele alınacaktır. III. bölümde, kullanılan veri setinin yapısı ve içeriği detaylı bir şekilde açıklanacaktır. IV. bölümde, önerilen yöntemler ve bu yöntemlerin teorik temelleri sunulacaktır. V. bölümde, veri seti üzerinde uygulanan ön işleme adımları ve önerilen yöntemlerin deneysel sonuçları tartışılacaktır. VI. bölüm, makalede gerçekleştirilen çalışmaları özetleyecek ve sonuçları değerlendirecektir. Son olarak, VII. bölümde, bu çalışmanın gelecekte nasıl geliştirilebileceği ve yapılabilecek olası çalışmalar ele alınacaktır.

## II. İLGİLİ ÇALIŞMALAR

Satranç bulmacalarının zorluk seviyesini tahmin etmek için yaygın olarak kullanılan temel yöntemlerden biri Glicko-2 yöntemidir [3]. Glicko-2 yöntemi, satranç bulmacalarını belirli bir Elo seviyesinden başlatarak, kullanıcıların bulmacaları çözme başarılarına göre Elo seviyesini dinamik olarak günceller. Bu yöntem, bulmacaların zorluk seviyesini belirlemek için etkili bir çözüm sunmakla birlikte, kullanıcı girdisine yüksek derecede bağımlı olması nedeniyle bazı sınırlamalara sahiptir.

Satranç bulmaca zorluk seviyesini tahmin etmeye yönelik ilk yapay zeka çalışmalarından biri, Stoiljkovikj ve arkadaşları tarafından gerçekleştirilmiştir. Bu çalışmada, küçük bir veri kümesi üzerinde, bulmacalar kolay, orta ve zor olmak üzere üç sınıfa ayrılmıştır [4].

Satranç makinelerinin gelişimiyle birlikte, arama ağaçları gibi yöntemlerin satranç bulmacalarının zorluk derecelerini tahmin etmek için kullanıldığı çeşitli çalışmalar da literatüre kazandırılmıştır [5]. Ancak, yakın tarihe kadar yapay zeka ile satranç bulmaca zorluk seviyesini tahmin etmeye yönelik kapsamlı bir literatür bulunmamaktadır.

IEEE Büyük Veri Kupası, bu alanda önemli bir adım atılmasını sağlamış ve yarışma kapsamında çeşitli çözüm yöntemleri geliştirilmiştir. Yarışmada ilk üçe giren çalışmalar şu şekilde özetlenebilir:

- **bread emoji (1. sıra) [6]:** Bu çalışmada, satranç tahtası üzerindeki pozisyonları temsil etmek için Maia [7] ve Leela [8] modelleri temel alınarak embeddinger elde edilmiştir. Bu embeddingerin anlamlandırılması için CNN, RNN ve MLP katmanlarını içeren bir yapı tasarlanmıştır. Test veri kümesinde **104.5K** ortalama karesel hata elde ederek yarışmada birinci olmuştur.
- **anansch (2. sıra) [9]:** Leela [8] modelinden esinlenerek geliştirilen derin bir sinir ağı yapısı kullanılmıştır. Bu çalışma, ResNet ve CNN blokları içeren bir ağ mimarisine sahiptir. Test veri kümesinde **120.7K** ortalama karesel hata elde ederek yarışmada ikinci olmuştur.
- **ousou (3. sıra) [10]:** Bu çalışmada, satranç motorlarından çıkarılan özelliklere dayalı bir LightGBM modeli kullanılmıştır. Giriş özellikleri üç ana kategoriye ayrılmıştır: (1) Satranç motorlarından çıkarılan özellikler, (2) el ile üretilen (hand-crafted) özellikler ve (3) bir sinir ağı tarafından tahmin edilen zorluk derecelendirmeleri. Test veri kümesinde **123.1K** ortalama karesel hata elde ederek yarışmada üçüncü olmuştur.

Bu çalışmada önerilen sistem, graf tabanlı özellik mühendisliği ve CNN-GRU tabanlı özelleştirilmiş bir yapay sinir ağı içermesiyle, literatürdeki mevcut çalışmalardan ayrılmaktadır.

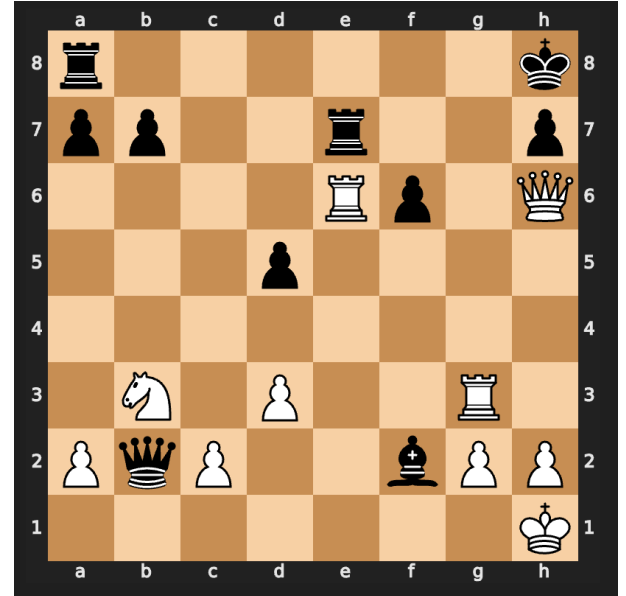
## III. VERİ KÜMESİ

IEEE Büyük Veri Kupası kapsamında paylaşılan veri kümesi, Tablo I'de detaylı olarak açıklanmıştır. Veri kümesi, her bir bulmaca için FEN bilgisini içermektedir. FEN, satranç tahtasındaki taşların konumunu tanımlamak için kullanılan

standart bir notasyondur ve satranç oyunlarının durumunu yeniden oluşturmada kritik bir rol oynar. Her FEN dizisi, mevcut oyun durumunu tam olarak tanımlayan altı boşlukla ayrılmış alandan oluşur.

Tablo I'de belirtilen FEN notasyonunun açıklaması aşağıdaki gibidir:

- **Taşların Konumu:** 'rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR', taşların oyunun başlangıç pozisyonlarını gösterir.
- **Aktif Renk:** 'w', sıranın beyazda olduğunu belirtir.
- **Rok Hakkı:** 'KQkq', her iki oyuncunun da şah ve vezir kanadında rok yapabileceği anlamına gelir.
- **Geçerken Alma Hedef Kare:** '-', geçerken almanın mümkün olmadığını ifade eder.
- **Yarım Hamle Saati:** '0', bir piyona hamle yapılmadığını veya taş alınmadığını, dolayısıyla saatin sıfırlanmadığını belirtir.
- **Tam Hamle Sayısı:** '1', oyunun ilk hamlesinin yapıldığını ifade eder.



Şekil 1. Örnek bir FEN notasyonunun görselleştirilmesi.

Şekil 1, veri kümesinden rastgele seçilen bir FEN notasyonunun görselleştirilmesini göstermektedir.

Bunun yanı sıra, veri kümesinde bulmacayı çözmek için yapılması gereken hamle dizisi PGN formatında paylaşılmıştır. PGN, satranç oyunlarının hamlelerini metin tabanlı bir formatta temsil etmek için kullanılan standart bir notasyondur. Ayrıca, Glicko-2 algoritması ile belirlenmiş olan ve bu çalışmada hedef değer olarak kullanılan bulmacanın zorluk seviyesi de veri kümesinde yer almaktadır. Zorluk seviyesiyle birlikte, bulmacanın deviasyonu, popüleritesi (kaç kere oynandığı) ve diğer metrikler de paylaşılmıştır. Bu ek bilgiler, zorluk seviyesinin daha iyi anlamlandırılmasını ve modelleme sürecinde kullanılmasını kolaylaştırmaktadır.

TABLO I  
VERİ KÜMESİ AÇIKLAMALARI

Alan Adı	Alan Açıklaması	Alan Türü	Örnek Değer
PuzzleId	Bulmaca kimliği	string	00sHx
FEN	Tahta pozisyonu standart notasyonda	string	q3k1nr/1pp1nQpp/3p4/1P2p3/4P3/B1PP1b2/B5PP/5K2 b k - 0 1 7
Moves	Bulmacanın çözümü PGN formatında	string	e8d7 a2e6 d7d8 f7f8
Rating	Bulmaca zorluk derecesi	int	1760
RatingDeviation	Zorluk seviyesindeki belirsizlik ölçüsü	int	85
Popularity	Beğenme ve beğenmeme farkı	int	10
NbPlays	Bulmacayı çözme deneme sayısı	int	350
Themes	Bulmaca temaları	string	2 hamlede mat
GameUrl	Bulmacanın kaynak bağlantısı	string	<a href="https://lichess.org/game1">https://lichess.org/game1</a>
OpeningTags	Bulmacanın çıktığı açılış	string	Sicilya Savunması

#### IV. YÖNTEM

Satranç bulmacalarının zorluk derecelerini tahmin etmek için iki temel yöntem geliştirilmiştir. İlk yöntem, satranç tahtasını özellik mühendisliği teknikleriyle modelleyerek, bu özellikleri ağaç tabanlı bir modele girdi olarak sağlamak üzerine kuruludur. İkinci yöntem ise, görsel özellikleri öğrenmek için bir CNN başı ve bağlamsal ilişkileri öğrenmek için bir GRU başı içeren özgün bir yapay sinir ağı tasarımı geliştirilmesini kapsamaktadır.

##### A. Ağaç Tabanlı Yöntem

Bu yöntemde, LightGBM algoritması kullanılmıştır. LightGBM, ağaç tabanlı bir makine öğrenimi algoritmasıdır ve özellikle büyük boyutlu veri kümelerinde yüksek performansı ve hızlı eğitim süresiyle öne çıkar. Bu algoritma, gradyan artırma tekniğini kullanarak bir dizi karar ağacını iteratif olarak optimize eder ve bu sayede karmaşık ilişkileri etkili bir şekilde modelleyebilir.

LightGBM algoritması, tabular bir girdi beklemektedir. Bu nedenle, satranç tahtasındaki pozisyonları temsil eden ham verilerin, LightGBM'in işleyebileceği bir tabular formata dönüştürülmesi gereklidir. Bu dönüşüm için iki farklı yöntem önerilmiştir.

1) *Lineer Yaklaşım*: Lineer yaklaşımda, tahtadaki her bir kare bir özellik olarak ifade edilmiş ve bu karelerde bulunan taşlar ASCII kodları ile temsil edilmiştir. Taşların ASCII kodları Tablo II'da verilmiştir. Bu yaklaşımla, satranç tahtası toplamda 64 özellikten oluşan bir vektör halinde modellenmiştir. Buna ek olarak, bulmacayı çözmek için tahaya uygulanması gereken hamleler dizisinin uzunluğu bir özellik olarak eklenmiş ve toplamda 65 özellikli bir veri seti oluşturulmuştur. Tablo III, bu dönüşüm sürecine ilişkin örnek bir veri setini göstermektedir.

2) *Çizge Tabanlı Yaklaşım*: Çizge Tabanlı Yaklaşımda her bir satranç tahtası için iki farklı çizge (*graph*) oluşturulmuştur: geçerli hamle çizgesi ve tehdit çizgesi. Çizge, düğümler (*nodes*) ve kenarlardan (*edges*) oluşan bir matematiksel yapı olarak ifade edilir. Bu yapının temel amacı, satranç tahtasının pozisyonel ilişkilerini modellemek ve bu ilişkilerden özellikler çıkarmaktır.

Bir çizge,  $G = (V, E)$  şeklinde tanımlanabilir. Burada:

- $V$ : Düğüm kümesi (*vertices*), satranç tahtasındaki kareleri ifade eder.

TABLO II  
TAŞ TIPLERİ VE ASCII KODLARI

Taş Tipi	Beyaz ASCII Kodu	Siyah ASCII Kodu
Piyon (P, p)	80	112
At (N, n)	78	110
Fil (B, b)	66	98
Kale (R, r)	82	114
Vezir (Q, q)	81	113
Şah (K, k)	75	107
Boş Kare (-)	45	45

TABLO III  
ÖRNEK TABULAR VERİ SETİ

Kare 1	Kare 2	Kare 3	Kare 4	...	Hamle Dizisi Uzunluğu
80	45	112	78	...	5
45	113	82	45	...	3
98	110	45	107	...	7

- $E$ : Kenar kümesi (*edges*), düğümler arasındaki ilişkileri temsil eder.

Geçerli hamle çizgesi,  $G_m = (V, E_m)$  şeklinde ifade edilir.

- $V$ : Satranç tahtasındaki 64 kareyi temsil eden düğümler kümesidir. Her düğümde, ilgili karedeki taşın tipi (örneğin piyon, at) ve rengi (beyaz veya siyah) tutulur.
- $E_m$ : İki düğüm arasında bir kenar, bir taşın mevcut pozisyonundan geçerli bir hamle ile başka bir kareye hareket edebileceği durumlarda tanımlanır.

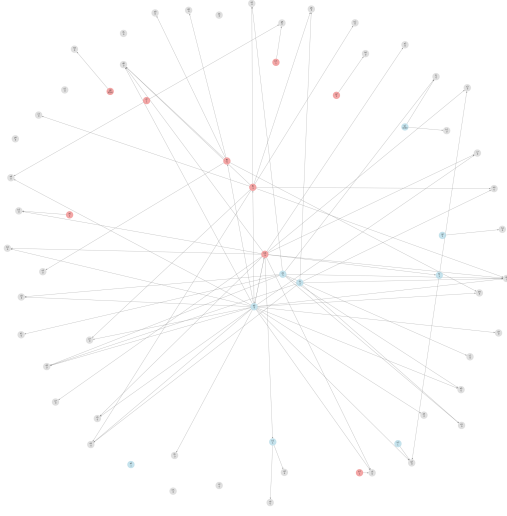
Bu çizge, taşların geçerli hamlelerini modelleyerek, satranç tahtasının mevcut durumu üzerinden oynanabilir hareketleri temsil eder.

Tehdit çizgesi,  $G_t = (V, E_t)$  şeklinde ifade edilir.

- $V$ : Aynı şekilde 64 kareyi temsil eden düğümler kümesidir.
- $E_t$ : İki düğüm arasında bir kenar, bir taşın bulunduğu kareden başka bir kareyi tehdit etmesi durumunda tanımlanır.

Bu çizge, taşların rakip taşları tehdit ettiği kareleri modellemek için kullanılır. Örneğin, bir vezirin çapraz veya dikey bir yönde herhangi bir taşı tehdit etmesi durumunda, tehdit edilen kare ile vezirin bulunduğu kare arasında bir kenar tanımlanır. Bu tür bir modelleme, satranç tahtasındaki stratejik pozisyonları ve tehdit dinamiklerini matematiksel bir yapıya

dönüştürerek analiz etmeyi mümkün kılar. Şekil 2, örnek bir tehdit çizgesini göstermektedir.



Şekil 2. Tehdit çizgesi örneği

Oluşturulan çizgeler kullanılarak, modelin pozisyonel bilgisini artırmak için çeşitli özellikler çıkarılmıştır. Bu özellikler, modelin tahtanın geometrik yapısını daha iyi anlamasına ve bu bilgiyi tahminleme sürecinde kullanmasına olanak sağlamaktadır. Aşağıda bu özelliklerden bazıları matematiksel olarak ifade edilmektedir. Burada  $\mathbb{K}(\cdot)$ , bir gösterge fonksiyonudur. Gösterge fonksiyonu, belirtilen koşul sağlanıyorsa 1, aksi takdirde 0 döndürür. Örneğin,  $\mathbb{K}(t \in \text{Merkez Kareler})$ ,  $t$  düğümünün merkez karelerden biri olması durumunda 1 döner; aksi takdirde 0 döner.

- 1) **Merkez Hakimiyeti:** Taşların merkez karelere ( $\{d4, e4, d5, e5\}$ ) olan etkisini ölçer.

$$C(c) = \sum_{(s,t) \in E} \mathbb{K}(t \in \text{Merkez Kareler} \wedge \text{Renk}(s) = c) \quad (1)$$

Burada  $C(c)$ ,  $c$  rengindeki taşların merkez kareleri kontrol etme sayısını;  $s$  ve  $t$ , sırasıyla kaynak ve hedef kareleri;  $\mathbb{K}$ , bir göstergesi fonksiyonunu ifade eder. Satrançta merkez hakimiyeti pozisyonel açıdan büyük avantajlar sağlayabildiği için model için kritik bir özelliktir.

- 2) **Tehdit Özellikleri:** Bir taşın rakip taşları tehdit ettiği kare sayısı ve tehdit edilen taşların toplam değerini ölçer.

$$T_{\text{count}}(c) = \sum_{(s,t) \in E} \mathbb{K}(\text{Renk}(t) \neq c) \quad (2)$$

$$T_{\text{value}}(c) = \sum_{(s,t) \in E} \text{Değer}(t) \cdot \mathbb{K}(\text{Renk}(t) \neq c) \quad (3)$$

Burada  $T_{\text{count}}(c)$ ,  $c$  rengindeki taşların tehdit ettiği karelerin sayısını;  $T_{\text{value}}(c)$  ise tehdit edilen taşların

toplam değerini ifade eder. Satrançta bir taşın rakibini tehdit edebilmesi, oyunun kontrolünü elinde tutmak açısından kritik bir öneme sahiptir. Bu özellik, modelin tahtadaki taşların tehdit potansiyelini anlamasına yardımcı olur.

- 3) **Hareket Kabiliyeti:** Bir taşın yapabileceği toplam geçerli hamle sayısını ifade eder.

$$M(c) = \sum_{v \in V_c} \text{OutDegree}(v) \quad (4)$$

Burada  $M(c)$ ,  $c$  rengindeki taşların toplam hareket kabiliyetini ve  $\text{OutDegree}(v)$ , düğüm  $v$  için çıkış derece sayısını ifade eder. Hareket kabiliyeti, modelin pozisyonel olarak oyunun açık mı yoksa kapalı mı olduğunu değerlendirmesine olanak sağlar. Ayrıca, hangi rengin daha avantajlı bir pozisyonda olduğunu belirlemek açısından da kritik bir özelliktir.

- 4) **Malzeme Üstünlüğü:** Her iki tarafın taşlarının değerleri arasındaki farkı ölçer.

$$\text{MaterialDiff} = \sum_{v \in V_{\text{white}}} \text{Değer}(v) - \sum_{v \in V_{\text{black}}} \text{Değer}(v) \quad (5)$$

Burada  $\text{MaterialDiff}$ , beyaz ve siyah taşlar arasındaki toplam malzeme farkını ifade eder. Malzeme üstünlüğü, satrançta avantajı belirlemenin en temel ve yaygın kullanılan yöntemlerinden biridir. Taşların değerlerine dayalı olarak yapılan bu değerlendirme, oyunun stratejik dengesi hakkında bilgi verir.

- 5) **Şah Güvenliği:** Şahın çevresindeki tehdit edilen karelerin sayısını ( $T_k$ ) ve güvenli karelerin sayısını ( $S_k$ ) ifade eder.

$$T_k = \sum_{v \in \text{KingAdjSquares}} \mathbb{K}(\text{Tehdit}(v)) \quad (6)$$

$$S_k = |\text{KingAdjSquares}| - T_k \quad (7)$$

Burada  $\text{KingAdjSquares}$ , şahın çevresindeki kareler kümesini ifade eder. Satrançta temel amaç, şahı güvende tutmaktır; bu nedenle, şah güvenliği özelliği, pozisyonel değerlendirmede önemli bilgiler sağlar. Bu özellik, şahın çevresindeki tehdit edilen karelerin ve güvenli karelerin analizini yaparak modelin savunma ve saldırı dengelerini anlamasına yardımcı olur.

- 6) **Bağlanabilirlik:** Çizgenin kümeleşme katsayısının, beyaz ve siyah taşlar için ayrı ayrı ortalamasını hesaplar.

$$C_{\text{avg}}(c) = \frac{1}{|V_c|} \sum_{v \in V_c} C(v) \quad (8)$$

Burada  $C_{\text{avg}}(c)$ ,  $c$  rengindeki taşlar için ortalama kümeleşme katsayısını ifade eder ve  $C(v)$  düğüm  $v$  için kümeleşme katsayısını belirtir. Taşların bir-biriyle uyum içinde hareket ederek rakibi tehdit etmesi, rakibin savunma stratejilerini zorlaştırır. Bu nedenle, bağlanabilirlik özelliği pozisyonel açıdan kritik bilgiler sağlayarak modelin tahtadaki taşların iş birliğini anlamasına yardımcı olur.

- 7) **Çözüm için hareket etmesi gereken ilk taş:** İlk hareket etmesi gereken taşı tespit eder.

$$T = f(s_1) \quad (9)$$

Burada  $T$ , ilk hamlede hareket eden taşı ifade eder;  $s_1$ , ilk hamlenin başlangıç karesidir ve  $f(s_1)$ , bu karede bulunan taşın türünü belirten bir fonksiyondur.

Satranç bulmacalarında çözüm için ilk hangi taşın hareket etmesi gerektiğini belirlemek bazı durumlarda oldukça zorlayıcı olabilir. Modelin, çözüm için hareket etmesi gereken taşı ayrı bir özellik olarak girdi alması, bulmacanın pozisyonel bilgisi hakkındaki anlayışını artırarak model performansını olumlu yönde etkiler.

Üretilen bu özellikler, Tablo III'deki mevcut özelliklerle birleştirilerek, LightGBM modelinin eğitimi için kullanılmıştır.

### B. Özelleştirilmiş Sinir Ağı Yöntemi

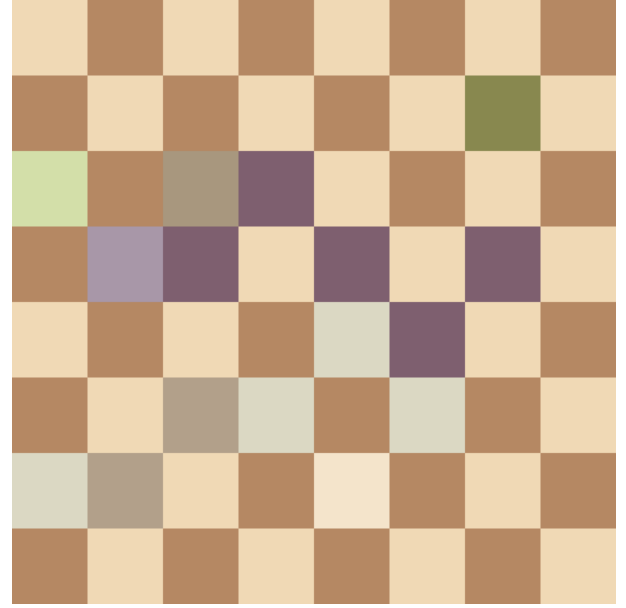
Çizgelerden elde edilebilecek anlamlı özellikler sınırlı olduğundan, satranç tahtasındaki pozisyonel bilgileri doğrudan öğrenebilen ve bulmacanın çözümü için gerekli hamlelerin oluşturduğu bağlamsal ilişkileri anlayabilen bir özelleştirilmiş sinir ağı tasarlanmıştır. Bu sinir ağı, pozisyonel ve bağlamsal bilgileri aynı anda işleyerek daha kapsamlı öğrenim sağlamayı hedeflemektedir.

Tahtanın pozisyonel özelliklerini öğrenmek için CNN modeli tercih edilmiştir. Özelleştirilmiş sinir ağının bir başında yer alan CNN modeli, satranç tahtasını giriş olarak alır. CNN modellerinin, resim işleme alanında yüksek performans gösterdiği birçok çalışmada kanıtlanmıştır. Satranç tahtasını CNN modeline uygun bir şekilde giriş olarak vermek için üç farklı yöntem önerilmektedir:

- 1) **Siyah-Beyaz Gösterim:** Tahtadaki kareler ve taşlar siyah-beyaz bir görüntü formatında temsil edilir. Bu yöntem en temel yaklaşımdır.
- 2) **RGB Gösterim:** Tahta, taşların değerlerini renklerle ifade eden bir RGB görüntüsü olarak temsil edilir. Şekil 3 bu yöntemi görselleştirmektedir.
- 3) **One-Hot Gösterim:** Her bir taş tipi ve renk kombinasyonu (toplamda 12 kanal) ile boş kareler için ek bir kanal kullanılarak 13 kanallı bir one-hot encoded gösterim oluşturulur. Bu yöntem, taşların pozisyonel ve türsel bilgilerini daha etkili bir şekilde öğrenmek için en mantıklı yaklaşımdır. Şekil 4 bu yöntemi görselleştirmektedir.

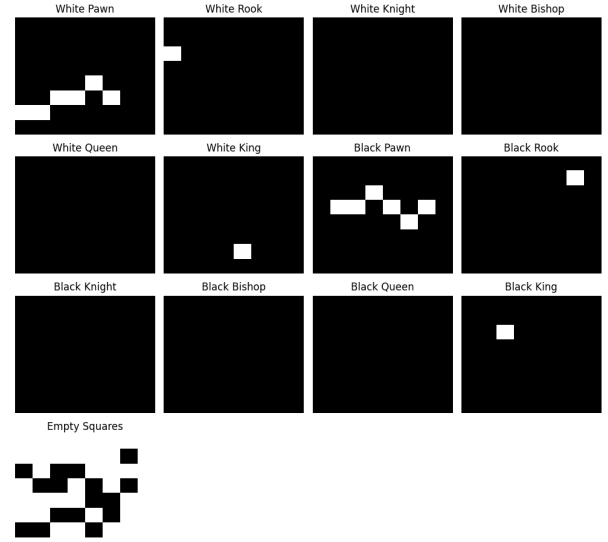
Bulmacayı çözmek için gerekli hamlelerin anlamlandırılması için GRU modeli kullanılmıştır. GRU, zamansal bilgiyi hafızasında tutabilme özelliği sayesinde uzamsal ve ardışık özelliklerin öğrenilmesini mümkün kılmaktadır. Satranç bulmacası çözerken uzamsal bilgi büyük önem taşımaktadır çünkü bir hamlenin etkisi genellikle sonraki hamlelerin stratejik bağlamını belirler.

GRU modeline girdi olarak, bulmacanın çözümü için yapılması gereken hamleler bir sıra dizisi halinde modellenmiştir. Bu işlem, satranç tahtasındaki karelerin Tablo IV'de gösterildiği gibi enüme edilerek dizesel bir



Şekil 3. RGB gösterimiyle satranç tahtası: Taş değerlerinin renklerle kodlanması

One-Hot Encoding Visualization for FEN: 8/6r1/R1kp4/1pp1p1p1/4Pp2/2PP1P2/PP2K3/8 b - - 0 42



Şekil 4. One-hot encoded satranç tahtası: 13 kanallı gösterim

forma dönüştürülmesiyle gerçekleştirilmiştir. Böylece her hamle, başlangıç ve hedef karelerin sayısal temsilleriyle ifade edilmiştir. Tablo V, bu yöntemeye dayalı olarak oluşturulan örnek bir hamle dizisini göstermektedir.

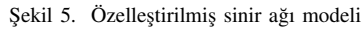
Özelleştirilmiş sinir ağı modeli, Şekil 5'de gösterilmiştir.

## V. DENEYSEL ÇALIŞMALAR

Bu bölümde, veri kümesinin anlamlandırılmasına yönelik gerçekleştirilen keşifsel veri analizi, bir önceki bölümde bahsedilen yöntemlerin değerlendirilmesi için oluşturulan test

	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>	<b>f</b>	<b>g</b>	<b>h</b>
8	56	57	58	59	60	61	62	63
7	48	49	50	51	52	53	54	55
6	40	41	42	43	44	45	46	47
5	32	33	34	35	36	37	38	39
4	24	25	26	27	28	29	30	31
3	16	17	18	19	20	21	22	23
2	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7

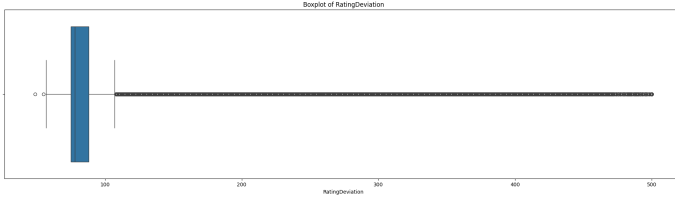
Hamle Sırası	Kaynak Kare	Hedef Kare
1	e2 (12)	e4 (28)
2	g8 (62)	f6 (45)



1) *Bulmaca zorluk derecesi:* Veri kümesinde toplamda 3.888.765 bulmaca paylaşılmıştır. Bu bulmacaların zorluk seviyelerinin ortalama değeri 1515'tir ve dağılımı Şekil 6'te gösterilmektedir.

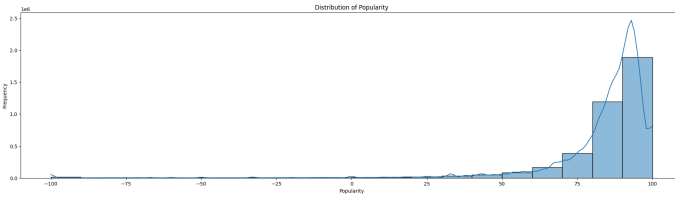
3) *Popülerite*: Bulmacaların popülerite dağılımı Şekil 9'de gösterilmektedir. Dağılım incelendiğinde, bulmacaların büyük çoğunluğunun yüksek popülerite değerlerine sahip olduğu, ancak bir kısmının 50'nin altında düşük popülerite değerlerine sahip olduğu dikkat çekmektedir.





Şekil 8. Bulmaca zorluk derecesi sapma değerlerinin IQR yöntemi ile temizlenmesi.

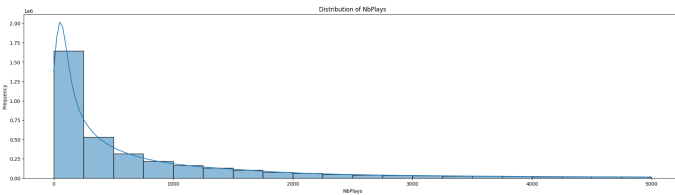
Düşük popülariteye sahip bulmacalar, kullanıcılar tarafından beğenilmedikleri için zorluk düzeyleri açısından güvenilir olmayabilir. Bu nedenle, eğitim kümesinden 65.5'in altındaki popülarite değerine sahip veriler çıkarılmıştır. Bu işlem, modelin yalnızca güvenilir zorluk seviyelerine dayalı olarak eğitilmesini sağlamak ve eğitim sürecindeki potansiyel gürültüyü azaltmak amacıyla gerçekleştirilmiştir.



Şekil 9. Bulmaca popülaritelerinin dağılımı.

4) *Bulmaca Oynanma Sayısı*: Bulmacaların oynanma sayılarının dağılımı Şekil 10'da gösterilmektedir. Dağılım incelendiğinde, oynanma sayılarının büyük ölçüde düşük değerlere yoğunlaştığı, ancak bazı bulmacaların çok yüksek oynanma sayılarına sahip olduğu görülmektedir. Bulmacaların ortalama oynanma sayısı 1674'tür.

Glicko-2 algoritması, kullanıcı girdisine doğrudan bağlı olduğu için oynanma sayısı düşük olan bulmacaların zorluk düzeyleri güvenilir olmayabilir. Özellikle, oynanma sayısı 10'dan küçük olan bulmacalar, yetersiz kullanıcı geri bildirimi nedeniyle model eğitimi sırasında hatalı yönlendirmelere neden olma potansiyeline sahiptir. Bu nedenle, eğitim kümesinden oynanma sayısı 10'dan küçük olan bulmacalar çıkartılmıştır. Bu işlem, modelin yalnızca yeterli kullanıcı geri bildirimine dayalı zorluk seviyeleri üzerinden eğitilmesini sağlamak amacıyla gerçekleştirilmiştir.



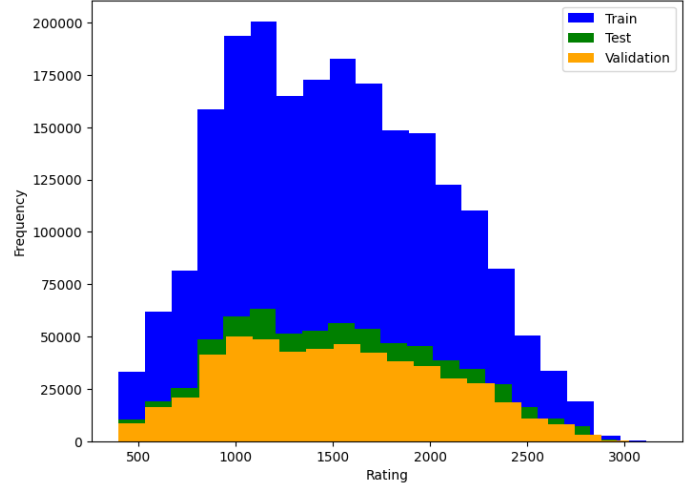
Şekil 10. Bulmacaların oynanma sayılarının dağılımı.

## B. Deneylerin Tasarlanması

Bölüm III'te önerilen yöntemlerin tutarlı bir şekilde değerlendirilebilmesi için veri kümesi, aynı dağılıma sahip

eğitim, doğrulama (validation) ve test kümelerine ayrılmıştır. Bu işlem, her bir kümede yer alan verilerin temsil gücünü artırarak modelin performansını objektif bir şekilde değerlendirmeyi amaçlamaktadır. Şekil 11, bu kümelere ayrılmış verilerin zorluk derecesi dağılımlarını göstermektedir.

Eğitim kümesi, modelin öğrenme süreci için kullanılırken, doğrulama kümesi, hiperparametre optimizasyonu ve model seçimi için kullanılmıştır. Test kümesi ise modelin genelleştirme yeteneğini değerlendirmek amacıyla ayrılmıştır. Kümeler arasında aynı dağılımın korunması, model performansının tutarlı bir şekilde karşılaştırılmasını sağlamaktadır.



Şekil 11. Eğitim, doğrulama ve test kümelerinin zorluk derecesi dağılımları.

## C. Deneyler

Deneylerde, LightGBM modeli hem lineer yaklaşım hem de çizge tabanlı yaklaşım için kullanılmış ve hiperparametre optimizasyonu Optuna kütüphanesi ile gerçekleştirilmiştir. Bu süreçte, her iki yaklaşım için de en iyi hiperparametreler optimize edilmiştir ve Tablo VI'da gösterilmektedir.

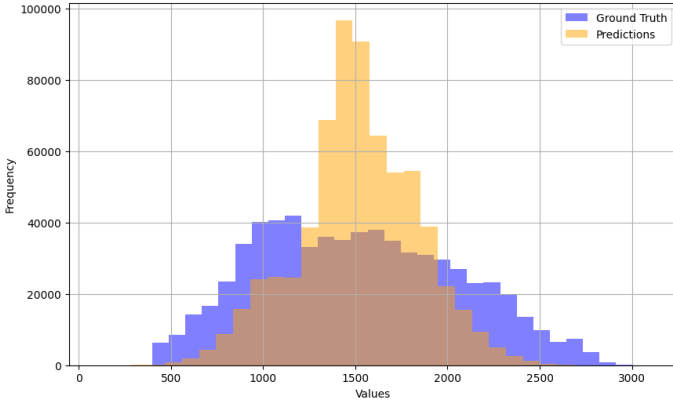
TABLO VI  
LIGHTGBM HIPERPARAMETRE OPTİMİZASYON SONUÇLARI.

Parametre	Lineer Yaklaşım	Çizge Tabanlı Yaklaşım
Objective	regression	regression
Metric	mse	mse
Boosting Type	gbdt	gbdt
Learning Rate	0.099980	0.099888
Num Leaves	148	148
Bagging Fraction	0.8631	0.7859
Bagging Frequency	6	1
Feature Fraction	1	1
Verbose	1	1

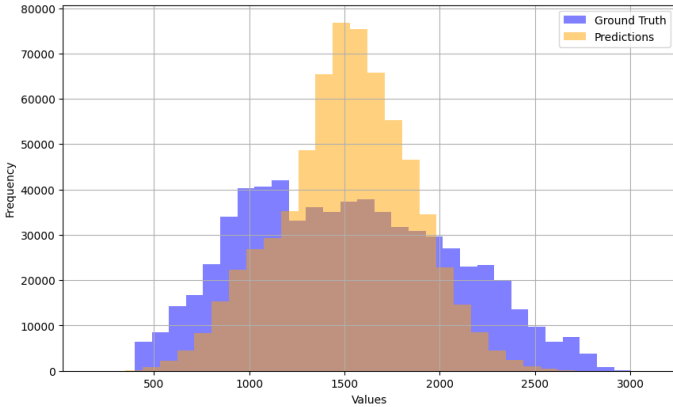
Hiperparametre optimizasyonu, modelin performansını artırmak ve her iki yaklaşımın veri kümesi üzerindeki başarımını adil bir şekilde karşılaştırmak amacıyla yapılmıştır. Bu süreçte, metrik olarak ortalama karesel hata (Mean Squared Error, MSE) kullanılmış ve en iyi sonuçları sağlayan parametreler belirlenmiştir.

Elde edilen parametrelerle model eğitilmiş ve test veri kümesi üzerinde sonuçları analiz edilmiştir. Şekil 12 ve Şekil 13, sırasıyla lineer yaklaşım ve çizge tabanlı yaklaşım ile eğitilen LightGBM modelinin test veri kümesi üzerindeki performansını göstermektedir.

Lineer yaklaşım için test veri kümesinde ortalama karesel hata (MSE) 176159 olarak elde edilmiştir. Buna karşılık, çizge tabanlı yaklaşım daha düşük bir hata ile 163668 MSE değerine ulaşmıştır. Bu sonuç, çizge tabanlı yaklaşım ile elde edilen özelliklerin modelin performansını artırmada önemli bir katkı sağladığını göstermektedir.



Şekil 12. Lineer yaklaşım için test veri kümesi üzerindeki performans.



Şekil 13. Çizge tabanlı yaklaşım için test veri kümesi üzerindeki performans.

Şekil 12, lineer yaklaşım ile geliştirilen modelin tahminlerinin, genellikle ortalama zorluk seviyesi etrafında yoğunlaştığını göstermektedir. Bu durum, tahtanın lineerleştirilmesinin ağaç tabanlı model için yeterince anlamlı özellikler üretmediğini ve modelin bu nedenle daha fazla genelleme yaptığını ortaya koymaktadır.

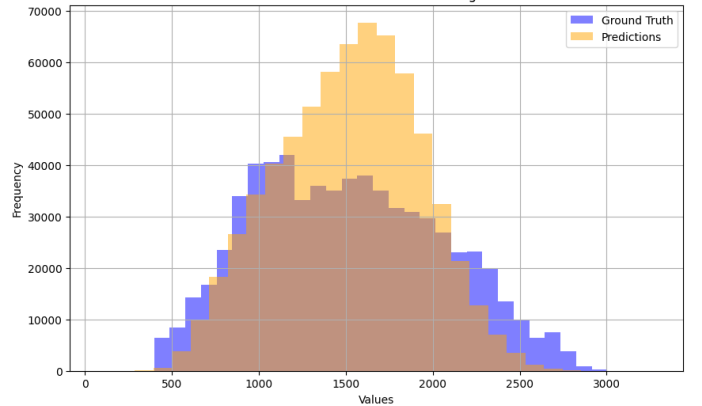
Şekil 13 ise çizge tabanlı yaklaşım ile geliştirilen özelliklerin, modelin bulmacaların özünü daha iyi kavramasını sağladığını göstermektedir. Bu yaklaşım, modelin zorluk seviyelerine göre daha dengeli tahminler yapmasına olanak tanımıştır. Özellikle 1000 zorluk seviyesi civarındaki bulmacalarda çizge tabanlı yaklaşımın performansının belirgin

şekilde iyileştiği görülmektedir. Ancak, her iki yaklaşım da 2500 üzeri zorluk seviyelerini tahmin etmede düşük performans sergilemektedir. Bu sonuç, bu tür yüksek zorluk seviyelerinin veri kümesindeki sınırlı temsiliyeti ile ilişkilendirilebilir.

Özelleştirilmiş yapay sinir ağı yönteminde kullanılan modelin parametreleri Tablo VII'de gösterilmiştir. Modelin parametre sayısının fazla olması nedeniyle hiper parametre optimizasyonu gerçekleştirilememiştir. Bunun yerine, Tablo VIII'te belirtilen hiper parametreler kullanılarak eğitim gerçekleştirilmiştir.

Modelin yapısı, satranç tahtasındaki pozisyonel bilgileri ve hareket dizilerini etkili bir şekilde öğrenmeyi amaçlamaktadır. Özellikle Conv2D katmanları, tahtanın görsel temsillerini öğrenmek için kullanılırken, GRU katmanı hareket dizilerinin zamansal ve bağlamsal ilişkilerini öğrenmede kritik bir rol oynamaktadır.

Şekil 14, eğitilen modelin test kümesi üzerindeki performansını göstermektedir. Özelleştirilmiş yapay sinir ağı, düşük zorluk düzeyindeki verileri yüksek doğruluk oranında tahmin ederek test MSE değerini 142150 seviyesine düşürmüştür. Bu başarı, modelin konvolüsyonel katmanlarının tahtadaki pozisyonel bilgiyi etkili bir şekilde öğrenmesinden ve GRU'nun hareket dizilerindeki bağlamsal ilişkileri yakalamasından kaynaklanmaktadır. Ancak, modelin yüksek zorluk seviyesindeki bulmacaları tahmin etmede başarısız olduğu görülmektedir. Bu durum, yüksek zorluk seviyelerinde daha karmaşık bir model yapısına ihtiyaç duyulabileceğini göstermektedir.



Şekil 14. Özelleştirilmiş yapay sinir ağı yaklaşımı için test veri kümesi üzerindeki performans.

Tablo IX üç farklı yaklaşımla elde edilen test MSE değerlerini göstermektedir. Özelleştirilmiş sinir ağı yöntemi diğer iki yöntemden çok daha iyi performans göstererek literatürdeki çözümlere en yakın MSE değerini çıktı olarak vermiştir.

## VI. SONUÇLAR

Bu çalışmada, IEEE Büyük Veri Kupası kapsamında paylaşılan satranç bulmacaları veri kümesi kullanılarak, bulmacaların zorluk seviyelerini tahmin edebilen bir model geliştirilmiştir. Modelin girdi olarak FEN ile ifade edilen



TABLO VII  
ÖZELLEŞTİRİLMİŞ YAPAY SINIR AĞI YAPISI

Katman	Giriş Boyutu	Çıkış Boyutu	Açıklama
Conv2d (fen_conv1)	(13, 8, 8)	(32, 8, 8)	3x3 konvolüsyon, 32 filtre, stride=1, padding=1
Conv2d (fen_conv2)	(32, 8, 8)	(64, 8, 8)	3x3 konvolüsyon, 64 filtre, stride=1, padding=1
MaxPool2d (fen_pool)	(64, 8, 8)	(64, 4, 4)	2x2 maksimum havuzlama (pooling), stride=2
Embedding (move_embedding)	(batch_size, sequence_length)	(batch_size, sequence_length, 16)	Hamle dizisi için embedding, boyut: 16
GRU (move_gru)	(batch_size, sequence_length, 16)	(batch_size, 32)	GRU, gizli boyut: 32
Linear (fc1)	(64*4*4 + 32 = 1056)	128	64 filtreli havuzlanmış özellikler + GRU çıktılarının bağlantılı katmanı
Linear (fc2)	128	64	Orta katman
Linear (fc3)	64	1	Çıkış katmanı, tahmini zorluk derecesi

TABLO VIII  
MODEL EĞİTİM PARAMETRELERİ

Parametre	Değer
Learning Rate	0.0001
Batch Size	512
Epoch Sayısı	20

TABLO IX  
MODELLERİN PERFORMANS KARŞILAŞTIRMASI

Model	Test MSE
Lineer Yaklaşım (LightGBM)	176159
Çizge Tabanlı Yaklaşım	163668
Özelleştirilmiş Yapay Sinir Ağı	142150

satranç tahtası durumu ve PGN ile belirtilen çözüm hamlelerini aldığı; çıktı olarak ise bulmacanın zorluk seviyesini tahmin ettiği bir yapı tasarlanmıştır.

Model geliştirme sürecinde üç farklı yöntem önerilmiştir. İlk yöntemde, satranç tahtası lineerleştirilerek her bir kare, model için bir özellik olarak ifade edilmiş ve bu özellikler LightGBM algoritması ile işlenmiştir. İkinci yöntemde, her bir bulmaca için iki çizge (geçerli hamle çizgesi ve tehdit çizgesi) oluşturulmuştur. Bu çizgeler üzerinden özellik mühendisliği gerçekleştirilmiş ve model performansında iyileştirmeler gözlemlenmiştir. Ancak, bu yöntemlerin performans iyileştirme potansiyeli sınırlı kalmıştır.

Bu sınırlamaları aşmak amacıyla özelleştirilmiş bir sinir ağı modeli geliştirilmiştir. Önerilen model, iki başlı bir yapıya sahiptir: birinci başlık, pozisyonel bilgileri öğrenmek için CNN tabanlıdır; ikinci başlık ise uzamsal ilişkileri anlamak için GRU tabanlıdır. Bu yöntemle, diğer iki yönteme kıyasla daha düşük hata değerleri elde edilmiş ve modelin performansı literatürdeki çalışmalara yaklaşmıştır.

Ayrıca, veri kümesi üzerinde gerçekleştirilen keşifsel veri analizi ile gürültülü veriler temizlenmiş ve bu sayede modelin tahmin doğruluğu artırılmıştır. Elde edilen sonuçlar, önerilen modelin satranç bulmacalarının zorluk seviyesini tahmin etme konusunda etkili bir yöntem olduğunu göstermektedir.

## VII. GELECEK ÇALIŞMALAR

Bu çalışmada önerilen ve diğer yöntemlere kıyasla daha iyi sonuçlar sağlayan özelleştirilmiş sinir ağı modelinin başarımını artırmak için çeşitli geliştirmeler yapılabilir. İlk olarak, veri kümesi zenginleştirilerek her hamle sonrası oluşan satranç tahtası pozisyonlarının modele dahil edilmesi

sağlanabilir. Bu zenginleştirme, özellikle CNN kısmına hamlelerin pozisyonel etkisini daha net bir şekilde kavrayabilme imkanı tanıyacaktı. Böylece, model hem pozisyonel hem de uzamsal bilgileri daha etkin bir şekilde öğrenebilir.

Bunun yanı sıra, özelleştirilmiş sinir ağı yöntemi üzerinde daha güçlü bir donanım kullanılarak hiperparametre optimizasyonu gerçekleştirilebilir. Bu optimizasyon, modelin performansında önemli iyileştirmeler sağlayabilir.

Ek olarak, satranç motorları kullanılarak her pozisyonun değerlendirilmesi veri kümesine eklenebilir. Bu yaklaşım, modelin pozisyonel olarak hangi tarafın avantajlı olduğunu daha iyi anlamasına yardımcı olabilir. Böylelikle, modelin zorluk seviyesi tahminlerinde daha yüksek bir doğruluk oranı yakalaması mümkün olacaktır.

Son olarak, CNN kısmında önceden eğitilmiş (pre-trained) özellik çıkarıcı ağlar kullanılarak, satranç tahtasının özelliklerini daha etkin bir şekilde öğrenebilen bir model oluşturulabilir. Bu tür bir yaklaşım, mevcut modelin öğrenme kapasitesini artırarak özellikle pozisyonel bilgilerin daha derin bir şekilde anlamlandırılmasına olanak tanıyacaktı.

## REFERANSLAR

- [1] H. A. Davidson, *A Short History of Chess*, Crown, 2012. ISBN: 9780307828293. Available at: <https://books.google.com.tr/books?id=PQBpr6KAdHkC>.
- [2] Claude E. Shannon, "A Chess-Playing Machine," in *Computer Games I*, edited by David N. L. Levy, Springer New York, New York, NY, 1988, pp. 81–88. Available at: [https://doi.org/10.1007/978-1-4613-8716-9\\_6](https://doi.org/10.1007/978-1-4613-8716-9_6). DOI: 10.1007/978-1-4613-8716-9\_6. ISBN: 978-1-4613-8716-9.
- [3] Mark E. Glickman, "The Glicko System," Boston University, vol. 16, no. 8, pp. 9, 1995.
- [4] S. Stoiljković, I. Bratko, M. Guid, and F. UNI, "A Computational Model for Estimating the Difficulty of Chess Problems," in *Proceedings of the Third Annual Conference on Advances in Cognitive Systems (ACS 2015)*, 2015, p. 7.
- [5] M. Guid and I. Bratko, "Search-based Estimation of Problem Difficulty for Humans," in *Artificial Intelligence in Education: 16th International Conference, AIED 2013, Memphis, TN, USA, July 9-13, 2013. Proceedings 16*, Springer, 2013, pp. 860–863.
- [6] T. Woodruff, O. Filatov, and M. Cagnetta, "The bread emoji Team's Submission to the IEEE BigData 2024 Cup: Predicting Chess Puzzle Difficulty Challenge," in *IEEE International Conference on Big Data, Big Data 2024*, Washington DC, USA, December 15–18, 2024. IEEE, 2024.
- [7] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," 2017. [Online]. Available: <https://arxiv.org/abs/1712.01815>
- [8] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson, "Aligning Superhuman AI with Human Behavior: Chess as a Model System," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1677–1687.

- [9] A. Schütt, T. Huber, and E. André, “Estimating Chess Puzzle Difficulty Without Past Game Records Using a Human Problem-Solving Inspired Neural Network Architecture,” in *IEEE International Conference on Big Data, Big Data 2024, Washington DC, USA, December 15-18, 2024*, IEEE, 2024.
- [10] S. Björkqvist, “Estimating the Puzzlingness of Chess Puzzles,” in *IEEE International Conference on Big Data, Big Data 2024, Washington DC, USA, December 15-18, 2024*, IEEE, 2024.