

# Maç Sonucu Tahmini

Ensar Akbaş  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
ensar.akbas@gmail.com

**Özet—** Bu projede, futbol maç sonucu tahmin modelleri geliştirmek amacıyla kapsamlı bir veri seti oluşturulmuştur. Veri, Maçkolik sitesinden web scraping yöntemiyle toplanmış; Liverpool, Manchester City, Manchester United, Arsenal, Chelsea ve Tottenham takımlarına ait uzun dönemli maç istatistikleri derlenmiştir. Ham veriler, eksik değerlerin giderilmesi, korelasyon analizi ve görselleştirme çalışmalarıyla detaylı biçimde incelenmiş; özellik mühendisliği (feature engineering) teknikleriyle modellemeye uygun yeni değişkenler üretilmiştir. Bu güçlü veri temeli üzerine, geçmiş maçlara ait sıralı istatistiksel bilgiler ve takımlara özgü statik özellikleri dikkate alan zaman serisi tahmin modelleri inşa edilmiştir. Özellikle Transformer mimarisi üzerine kurulu Informer, Reformer, Autoformer, PatchTST ve TimesNet gibi derin öğrenme modelleri kullanılarak maç sonuçları tahmin edilmeye çalışılmıştır. Böylece proje, veri toplama ve ön işleme adımlarından başlayarak ileri düzey modelleme süreçlerine kadar uçtan uca bir makine öğrenimi yaklaşımı sunmaktadır.

**Keywords—**Web scraping, Feature engineering, Veri ön işleme, Transformer, Zaman serisi

## I. GİRİŞ

Futbol, yalnızca bir spor dalı olmanın ötesinde, büyük miktarda veri üreten ve bu verilerin analiziyle önemli içgörüler sunabilen bir alandır. Bu projede amaç, geçmiş futbol maçlarından elde edilen istatistiksel verileri kullanarak, gelecekteki maç sonuçlarını tahmin edebilecek nitelikte bir sistem geliştirmektir. Proje kapsamında, İngiltere Premier Lig'in önde gelen takımlarına ait geçmiş maç verileri Maçkolik platformundan web scraping yöntemiyle toplanmış; veri temizleme, ön işleme ve özellik mühendisliği süreçleri titizlikle uygulanarak modellemeye uygun bir yapı oluşturulmuştur. Oluşturulan bu veri yapısı yalnızca istatistiksel analizler için değil, aynı zamanda derin öğrenme tabanlı zaman serisi modelleri için de temel oluşturmuştur. Bu doğrultuda, geçmiş maç dizilerini ve takıma özgü sabit bilgileri birlikte işleyebilen Transformer tabanlı modern modellerle tahminleme yapılmıştır. Informer, Reformer, Autoformer, PatchTST ve TimesNet gibi mimariler kullanılarak, hem sıralı bilgilerin hem de yapısal örüntülerin etkili şekilde modellenmesi hedeflenmiştir. Böylece proje, veri toplama ve işleme sürecinden model eğitime kadar bütünsel bir öğrenme süreci sunmakta, futbol veri analitiği alanında uygulanabilir ve ölçeklenebilir bir çözüm önerisi getirmektedir.

## II. VERİ TOPLAMA

Veriler, Maçkolik web sitesinden Python programlama dili kullanılarak çekilmiştir. Kodlama işlemleri PyCharm editörü üzerinde yürütülmüş, verileri çekmek için Selenium kütüphanesi tercih edilmiştir. Ayrıca verilerin işlenmesi ve analizinde pandas, numpy ve görselleştirme için matplotlib kütüphanelerinden yararlanılmıştır. Süreç boyunca altı farklı takımın (Liverpool, Manchester City, Manchester United,

Chelsea, Arsenal, Tottenham) her biri için yaklaşık 700 maçlık veri toplanmıştır. Bu işlemler sırasında Google Chrome Driver kullanılmış ve veri çekimi sırasında Maçkolik platformu herhangi bir bot koruması veya erişim engeli uygulamadığı için süreç kesintisiz ve verimli bir şekilde tamamlanmıştır.

## III. VERİ SETİ

Projenin veri seti, 2013-2014 sezonu ile 2024-2025 sezonu arasındaki Premier Lig maçlarını kapsamaktadır. Liverpool, Manchester City, Manchester United, Chelsea, Arsenal ve Tottenham olmak üzere 6 farklı takım için oluşturulan bu veri setinde, her takım adına yaklaşık 700 maç verisi bulunmaktadır. Takımlara ait veriler bir excel tablosunda toplanmıştır. Tekrarlı veriler silinince 3700 adet veri elde edilmiştir. Bu kapsamda, toplam 31 özellik (feature) yer almaktadır. Özelliklerin bir bölümü doğrudan maç istatistiklerinden elde edilirken; bazıları ise özellik mühendisliği (feature engineering) süreçleriyle türetilmiştir. Bu yapı, takımların geçmiş performanslarını daha anlamlı şekilde modelleyebilmek için tasarlanmıştır.

### A. Features (Özellikler)

Tarih	Rakip ID	Takım ID
Is_Home	Sonuç	Gol
Rakip Gol	Topla Oynama(%)	Şut
İsabetli Şut	Başarılı Pas	Pas Başarısı(%)
Korner	Faul	Ofsayt
Rakip Topla Oynama (%)	Rakip Şut	Rakip İsabetli Şut
Rakip Başarılı Pas	Rakip Pas Başarısı(%)	Rakip Korner
Rakip Faul	Rakip Ofsayt	

Tablo 1. Maçkolik üzerinden çekilen özellikler

- Takım ID:** Takıma özgü kimlik numarasıdır. (Örneğin: Liverpool=1, Manchester City=2)
- Is\_Home:** Maçın ev sahibi olarak oynanıp oynanmadığını belirtir. (Ev Sahini=1, Deplasman=0)
- Sonuç:** Galibiyet: 1, Beraberlik: 0, Mağlubiyet: -1 olarak tutulmaktadır. Hedef değişkenimizdir.

### B. Feature Engineering (Özellik Mühendisliği)

Şut Verimliliği (İsabetli Şut/Şut)	Gol Son 5 Maçın Galibiyet Oranı	Sezon
Ay	Haftanın Günü	Son 5 Maçın Gol Ortalaması

Tablo 2. Feature engineering sonucu oluşan yeni özellikler

- **Sezon/Ay:** Maçın hangi yıl ve ayda yapıldığını ifade eder. (Örneğin: Sezon=2024, Ay=4)
- **Haftanın Günü:** Maçın hangi gün yapıldığını ifade eder. (Örneğin: 1=Pazartesi, 2=Salı)

Bu veri seti, sonraki aşamalarda analiz, görselleştirme ve makine öğrenmesi modelleri için temel kaynak olarak kullanılacaktır.

#### IV. VERİ ÖN İŞLEME VE TEMİZLEME

Toplanan ham veriler, analiz ve modelleme süreçlerinde kullanılmadan önce çeşitli temizleme ve ön işleme adımlarından geçirilmiştir. Bazı maçlarda istatistiklerin eksik olması nedeniyle, ilgili sütunlarda boş (*null*) değerler tespit edilmiştir. Bu durum özellikle "*Topla Oynama*", "*Pas Başarısı*", "*Başarılı Pas*" gibi istatistiklerde görülmüştür. Bu eksik verileri doldurmak amacıyla iki farklı yaklaşım kullanılmıştır:

**Sütun Ortalamasıyla Doldurma:** Eksik değerler, ilgili sütunun genel ortalaması ile doldurularak, verinin genel yapısı korunmuştur.

**Önceki Verilere Yakınlaştırma:** Bazı durumlarda, eksik değerler bir önceki veya benzer maçlardaki istatistikler göz önünde bulundurularak manuel olarak doldurulmuştur. Böylece ani sapmaların önüne geçilmiştir.

Ayrıca, verilerin bütünlüğünü korumak adına tarih biçimleri, takım ID'leri ve sonuç değerleri gibi temel alanlarda format düzeltmeleri ve tutarlılık kontrolleri yapılmıştır. Bu adımlar, modelin güvenilir sonuçlar üretebilmesi için temel bir hazırlık sürecidir.

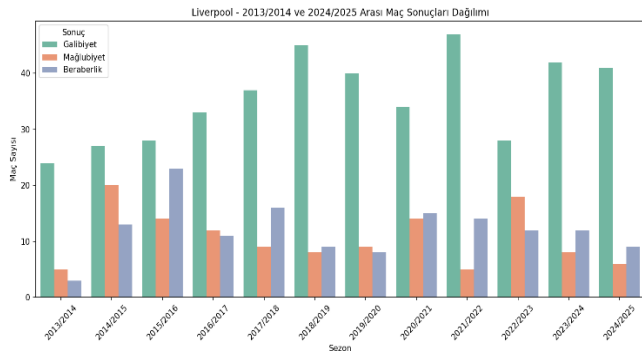
#### V. VERİ GÖRSELLEŞTİRME

Veri analizi sürecinde, istatistiksel içgörülerin daha anlaşılır ve yorumlanabilir hale gelmesi için veri görselleştirme önemli bir adımdır. Bu projede, görselleştirme işlemleri hem genel eğilimleri ortaya koymak hem de değişkenler arasındaki ilişkileri incelemek amacıyla gerçekleştirilmiştir. Özellikle Liverpool takımı üzerinde yapılan analizlerle örneklemeler sunulmuştur.

### Liverpool Üzerinden Görselleştirme Örnekleri

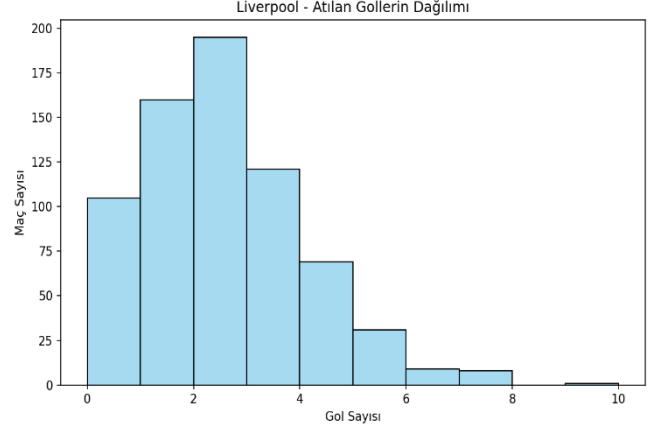
#### A. Sezonlara Göre Maç Sonuçları Dağılımı

Bu grafik ile her sezon boyunca kazanılan, kaybedilen ve berabere biten maçların dağılımı görselleştirilmiştir. Böylece takımın dönemsel performans dalgalanmaları net bir şekilde gözlemlenmiştir.



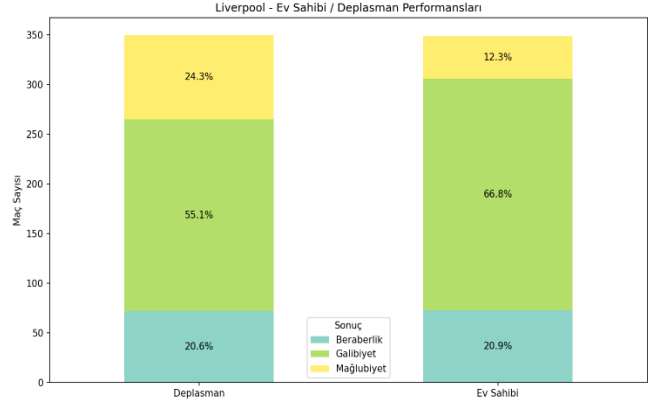
#### B. Atılan Gollerin Dağılımı

Takımın maç başına attığı gollerin frekansı incelenmiş, hangi gol aralığında en çok maç oynandığı grafiklerle gösterilmiştir.



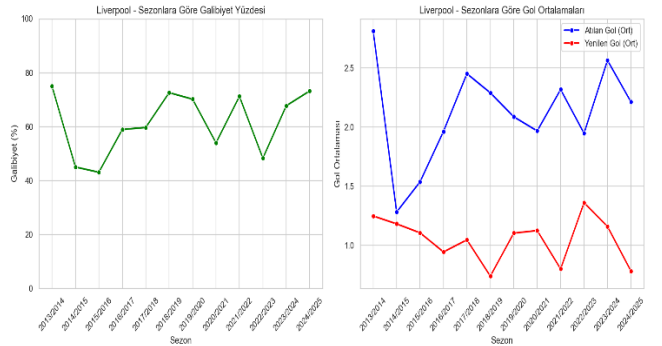
#### C. Ev Sahibi / Deplasman Performansları

Takımın iç saha ve dış saha maçlarındaki galibiyet oranları karşılaştırılarak performans farkları ortaya konmuştur.



#### D. Sezonlara Göre Galibiyet Yüzdesi ve Gol Ortalamaları

Her sezonun galibiyet yüzdesi ile maç başına düşen gol ortalaması birlikte incelenerek takımın genel başarımı değerlendirilmiştir.



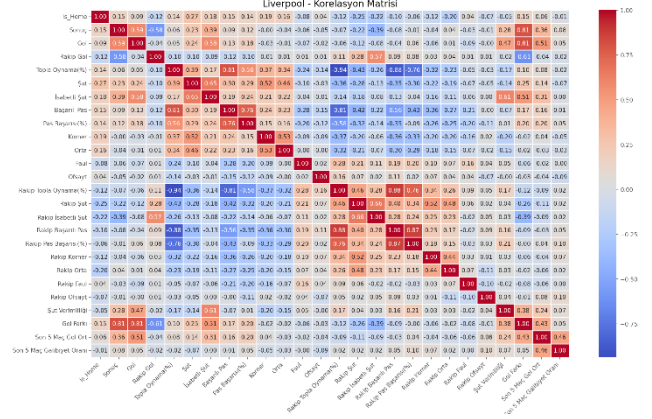
#### Korelasyon Analizi

Korelasyon analizi, iki değişken arasındaki doğrusal ilişkinin yönünü ve gücünü ölçen istatistiksel bir yöntemdir. Bu projede, maç istatistikleri arasındaki ilişkileri incelemek

amacıyla korelasyon analizinden yararlanılmış ve sonuçlar aşağıdaki başlıklar altında görselleştirilmiştir.

### E. Korelasyon Matrisi

Tüm sayısal değişkenler arasındaki korelasyon katsayıları ısı haritası ile gösterilerek, hangi değişkenlerin birbirini daha fazla etkilediği görsel olarak sunulmuştur.



### F. İsabetli Şut ve Gol İlişkisi

Orta düzeyde pozitif bir ilişki gözlemlenmiştir. Bu sonuç, daha fazla isabetli şut çeken takımların gol atma olasılığının önemli ölçüde arttığını göstermektedir.

### G. Topla Oynama ve Sonuç İlişkisi

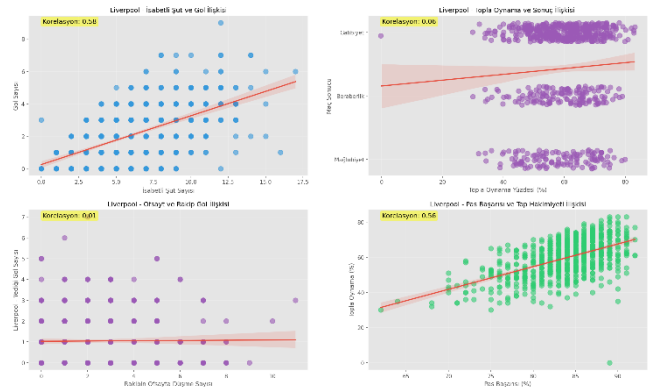
Bu oldukça düşük korelasyon, topa sahip olma yüzdesinin maç sonucunu doğrudan belirlemediğini, başka faktörlerin daha etkili olduğunu ortaya koymaktadır.

### H. Ofsayt ve Rakip Gol İlişkisi

Neredeyse sıfıra yakın bir ilişki bulunmuştur. Bu, hücumdaki ofsayt hatalarının savunma performansı veya rakip gol sayısı ile bağlantılı olmadığını göstermektedir.

### İ. Pas Başarısı ve Top Hakimiyeti

Orta düzeyde güçlü bir pozitif ilişki gözlemlenmiştir. Başarılı pas oranı arttıkça topa sahip olma yüzdesinin de arttığı açıkça görülmektedir.



Bu görselleştirmeler, hem istatistiksel değerlendirme hem de modelleme aşamasında önemli ipuçları sağlamış ve veriyle ilgili çıkarımların doğruluğunu artırmıştır.

## VI. MODELLEME

Bu bölümde, oluşturulan veri seti üzerinde uygulanan derin öğrenme modelleri ele alınmıştır. Projenin temel hedeflerinden biri, geçmiş maçlara ait sıralı istatistiksel veriler ve takımlara özgü sabit özellikler üzerinden gelecek maç sonucunu (galibiyet, beraberlik, mağlubiyet) tahmin edebilecek etkili bir model geliştirmektir. Bu doğrultuda, zaman serisi verilerini doğrudan modelleyebilen ve uzun bağımlılıkları öğrenmede başarılı olan Transformer tabanlı modern derin öğrenme mimarileri tercih edilmiştir. Modelleme sürecinde, hem sıralı (*sequence*) verilerin zaman içindeki örüntülerini yakalayabilen hem de statik (*static*) bilgileri değerlendirebilen yapılar kullanılmıştır. *Informer*, *Reformer*, *Autoformer*, *PatchTST* ve *TimesNet* gibi zaman serisi tahmininde son yıllarda öne çıkan modeller denenmiş; bu modellerin mimari farklılıkları, öğrenme kapasiteleri ve tahmin performansları detaylı olarak karşılaştırılmıştır. Modeller, eğitim ve test veri kümeleri üzerinde çeşitli metriklerle değerlendirilmiş, doğruluk, kayıp ve sınıf dengesine duyarlılık gibi performans ölçütleri dikkate alınmıştır. Bu bağlamda, izleyen alt bölümlerde her bir modelin yapısı, çalışma prensipleri ve proje kapsamındaki uygulama sonuçları ayrıntılı biçimde sunulmaktadır.

### Model Öncesi Hazırlık

Modelleme aşamasına geçmeden önce, sınıflandırma probleminin yapısına uygun olacak şekilde dikkatli bir veri hazırlık süreci yürütülmüştür. İlk olarak, yalnızca etiketlenmiş maçlar seçilmiş ve maç sonuçları  $\{-1, 0, 1\}$  biçiminden  $\{0: \text{mağlubiyet}, 1: \text{beraberlik}, 2: \text{galibiyet}\}$  olacak şekilde yeniden kodlanmıştır. Veri setinde, her maç örneği için iki temel veri türü hazırlanmıştır:

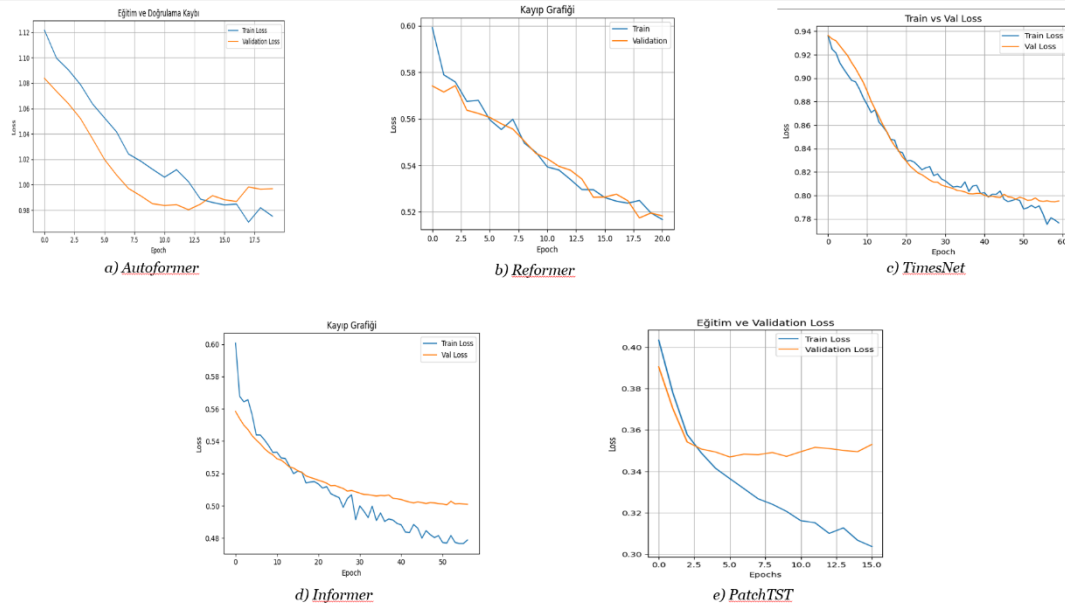
**Zaman serisi (sekans) özellikleri:** Hem takımın hem de rakibin geçmiş 10 maçına ait 20 adet istatistiksel özellik (şut, isabetli pas, korner, topa sahip olma yüzdesi vb.) toplanmıştır.

**Statik özellikler:** Takıma ve maça özgü değişmeyen bilgiler (ev sahibi olma durumu, sezon, ay, haftanın günü, şut verimliliği, son 5 maç ortalamaları) kullanılmıştır.

Zaman sırasına göre sıralanan veri, geçmiş maçları içerecek şekilde örneklendirilmiş ve sabit uzunlukta sekans matrisleri oluşturulmuştur. Statik verilerde kategorik sütunlar (sezon, ay vb.) one-hot encoding yöntemiyle dönüştürülürken, sayısal değişkenler standartlaştırılmıştır. Veri seti zaman sırasına bağlı kalacak şekilde %70 eğitim, %15 doğrulama, %15 test olarak ayrılmıştır. Her veri kümesindeki sınıf dağılımı ayrı ayrı analiz edilmiştir. Bu analiz sonucunda sınıf dengesizliğinin (özellikle beraberlik, mağlubiyet sınıflarının azlığı) model performansını olumsuz etkileyebileceği görülmüştür. Bu problemi hafifletmek için farklı stratejiler denenmiştir:

**Class weighting:** Kayıp fonksiyonuna sınıf ağırlıkları dahil edilerek azınlık sınıfların etkisi artırılmıştır.

**Focal loss:** Dengesiz sınıflarda modelin zorlandığı örnekler daha fazla ağırlık vererek öğrenmeyi iyileştiren bu kayıp fonksiyonu bazı modellerde başarıyla uygulanmıştır.



Görsel 1: Modellere ait epoch size - loss grafikleri

Parameters	Informer	Reformer	Autoformer	PatchTST	TimesNet
Epoch	57	21	19	15	60
Batch	32	32	32	32	32
Learning Rate	0.0001	0.0002	0.0002	0.0005	0.0001
Optimizer	Adam	Adam	Adam	Adam	Adam

Görsel 2: Modellerde kullanılan parametreler ve değerleri

#### A. Informer

Informer modeli, projede uygulanan zaman serisi tabanlı yaklaşımlar arasında en uzun eğitim döngüsüne sahip modelden biri olmuştur. Modelin eğitim sürecinde kullanılan düşük öğrenme oranı ve yüksek epoch sayısı, parametrelerin daha hassas bir şekilde ayarlanmasına olanak tanımış, bu da modelin genelleme kapasitesini olumlu etkilemiştir.

Eğitim süreci boyunca kayıp (loss) eğrileri dikkatlice gözlemlenmiş, Görsel 1' deki grafik doğrultusunda, eğitim ve doğrulama kayıplarının istikrarlı biçimde azaldığı görülmüştür. Özellikle doğrulama kaybının belirli bir noktadan sonra durağanlaşması, modelin yeni veriye karşı aşırı duyarlı hâle gelmeden öğrenmeyi sonlandırdığını göstermektedir. Eğitim ve doğrulama eğrileri arasındaki farkın makul seviyede kalması, overfitting riskinin başarılı bir şekilde yönetildiğine işaret etmektedir.

Modelin bu kararlı öğrenme süreci, geçmiş maçlardan elde edilen zaman serisi verileriyle uyumlu çalışabildiğini ve genel örüntüleri öğrenme konusunda başarılı olabileceğini göstermektedir.

#### B. Reformer

Reformer modeli, dikkat mekanizmasını daha verimli ve hafif bir yapıya indirgemesiyle bilinen bir zaman serisi modelidir. Bu model, projede uygulanan diğer modellere kıyasla daha az epoch ile eğitilmiş; düşük epoch sayısına rağmen kararlı bir öğrenme süreci geçirmiştir. Görsel 2'deki parametrelerden görülebileceği üzere, nispeten yüksek bir öğrenme oranı tercih edilmiş ve eğitim süresi sınırlı tutulmuştur. Bu tercihler, modelin hızlı öğrenmesini teşvik ederken, overfitting riskini de azaltmıştır.

Görsel 1 'deki kayıp grafiği incelendiğinde, hem eğitim hem de doğrulama kayıplarının ilk birkaç epoch içinde hızlıca azaldığı, sonrasında ise birbirine yakın ve paralel seyir izlediği gözlemlenmektedir. Doğrulama kaybında belirgin dalgalanmaların olmaması, modelin veriye karşı istikrarlı bir şekilde genelleme yapabileceğini göstermektedir. Ayrıca doğrulama eğrisinin eğitim eğrisiyle çok yakın seyretmesi, aşırı öğrenme eğiliminin başarılı şekilde kontrol edildiğine işaret etmektedir.

### C. Autoformer

Autoformer modelinde eğitim sürecinde kullanılan hiperparametreler dengeli bir yapı sunarken, modelin öğrenme süreci Görsel 1'deki kayıp grafiği üzerinden değerlendirildiğinde oldukça dikkat çekici bir profil sergilemiştir.

Modelin eğitim kaybı düzenli bir şekilde azalmış olsa da doğrulama kaybı zaman zaman dalgalı bir seyir izlemiştir. Bu dalgalanma, modelin bazı dönemlerde doğrulama verisine fazla uyum sağlamakta zorlandığını gösterebilir. Özellikle birkaç epoch boyunca doğrulama kaybının sabit kalması veya artış göstermesi, modelin genel örüntüleri öğrenirken bazı sınıflarda tutarsız tahminlerde bulunduğuna işaret edebilir.

Buna rağmen, eğitim ve doğrulama eğrileri genel olarak birbirine yakın seyrettiğinden, modelin overfitting eğilimi yüksek değildir. Öğrenme oranı, epoch sayısı ve optimizer gibi parametrelerin dengeli kullanılması, bu istikrarın sağlanmasında etkili olmuştur. Autoformer, yapısal karmaşıklığına rağmen kayıp değerlerini belirli bir seviyede tutmayı başarmış ve dengeli bir eğitim süreci ortaya koymuştur.

### D. PatchTST

PatchTST modeli, zaman serisi verisini yama (patch) temelli bir yaklaşımla işlemeye olanak tanıyan ve özellikle kısa vadeli örüntüleri etkili biçimde yakalamayı hedefleyen bir yapıya sahiptir. Modelin eğitim sürecinde orta seviyede bir epoch sayısı tercih edilmiş, bu da hem öğrenme süresini makul düzeyde tutmuş hem de modelin yeterince bilgi edinmesini sağlamıştır.

Görsel 1'de yer alan kayıp grafiği incelendiğinde, eğitim ve doğrulama kayıplarının ilk epoch'larda birlikte hızlıca azaldığı, ancak sonraki aşamalarda aralarında belirgin olmasa da ölçülü bir fark olduğu gözlemlenmektedir. Bu fark, modelin eğitim verisine kısmen daha fazla uyum sağladığını gösterse de, doğrulama kaybındaki genel eğilimin kararlı olması modelin aşırı öğrenmeye girmediğini düşündürmektedir. Özellikle doğrulama eğrisinin büyük dalgalanmalar göstermemesi, öğrenme sürecinin istikrarlı ilerlediğine işaret eder.

PatchTST'nin bu yapısı, modelin yüksek kapasitesine rağmen veri setine aşırı duyarlı hale gelmeden genellenebilir örüntüleri öğrenebildiğini göstermektedir. Eğitim süreci boyunca doğrulama kaybı eğitim kaybına yakın kalmasa da kontrol altında seyretmiş, bu da modelin az bir overfitting riski ile verimli biçimde eğitilebildiğini ortaya koymuştur.

### E. TimesNet

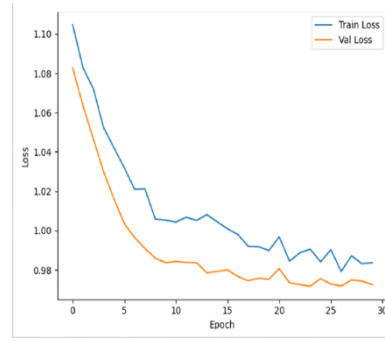
TimesNet modeli, zaman serisi verilerinde dönemsel ve ardışık örüntüleri yakalamaya yönelik özelleştirilmiş bir mimari sunar. Eğitim sürecinde, orta düzey bir epoch sayısı ve dengeli hiperparametre tercihleriyle modelin aşamalı bir öğrenme süreci yaşadığı gözlemlenmiştir.

Görsel 1'te yer alan kayıp eğrileri incelendiğinde, eğitim kaybının düzenli olarak azaldığı, doğrulama kaybının ise bir süre sabit kaldıktan sonra hızlı bir düşüş gösterdiği görülmektedir. Bu durum, modelin ilk aşamalarda doğrulama verisine genelleme yapmakta zorlandığını, ancak belirli bir

öğrenme eşiğini aştıktan sonra daha etkili sonuçlar ürettiğini göstermektedir. Doğrulama kaybındaki ani düşüşe rağmen, eğitim ve doğrulama eğrileri arasındaki fark kontrol altında kalmış; bu da modelin overfitting'e eğilim göstermediğini düşündürmektedir. Ayrıca doğrulama kaybının sonlara doğru yatay bir seyir izlemesi, modelin eğitim sürecinde doygunluğa ulaştığına işaret etmektedir. TimesNet, öğrenme sürecinde kısa vadede yavaş ilerlemiş olsa da uzun vadede istikrarlı ve dengeli bir performans ortaya koymuştur. Bu yapı, modelin karmaşık zaman serisi örüntülerini öğrenebilme potansiyelini göstermektedir.

Modellere ait diğer değerler (Karmaşıklık matris, ROC eğrileri, eğitim ve çıkarım süreleri, AUC, Sensitivity, Specifity metrikleri) için linke bakınız: [Link](#)

### F. GPT



Görsel 3: GPT modeline ait epoch size - loss grafiği ve performans metrikleri

Eğitim sürecine Görsel 3'e ait loss grafiği incelendiğinde, modelin ilk 10 epoch boyunca doğrulama kaybında belirgin bir iyileşme gösterdiği görülmektedir. Bu dönemde doğrulama doğruluğunda da ciddi bir artış gerçekleşmiş ve model 10. epoch'tan sonra daha durağan bir faza geçmiştir. Sonraki epoch'larda doğrulama kaybı ve doğruluğu küçük dalgalanmalarla birlikte sabit kalmış, bu da modelin erken aşamada dengesini bulduğunu göstermektedir. 30. epoch'ta erken durdurma uygulanarak eğitim sonlandırılmıştır.

Metriklere bakıldığında, modelin %48 civarında precision, recall ve F1-score değerlerine ulaştığı; bunun da sınıflar arasında dengeli bir performansa işaret ettiği söylenebilir. Ayrıca %56.07 doğruluk oranı ile GPT modeli, genel sınıflandırma başarısı açısından deneydeki en güçlü adaylardan biri olmuştur.

### VII. SONUÇ

Bu çalışmada kullanılan beş farklı zaman serisi modeli, geçmiş maç istatistiklerine dayalı olarak futbol maç sonucu tahmini yapma amacıyla değerlendirilmiştir. Her bir model için doğruluk (accuracy), hassasiyet (precision), duyarlılık (recall) ve F1 skoru gibi temel performans metrikleri hesaplanmış ve Görsel 4'de sunulmuştur.



Model	Precision	Recall	F-1 Score	Accuracy
Informer	0.4881	0.4357	0.3950	0.3911
Reformer	0.4700	0.3600	0.2800	0.2800
Autoformer	0.4900	0.4900	0.4700	0.5900
PatchTST	0.5000	0.5000	0.4900	0.5300
TimesNet	0.4500	0.4500	0.4400	0.4900

**Görsel 4:** Modellerle ait performans metrikleri

Karşılaştırma yapıldığında, en başarılı genel performansı Autoformer modeli sergilemiştir. %59 doğruluk oranı ve 0.47 F1 skoru ile hem sınıflandırma başarımı hem de istatistiksel denge açısından diğer modellere kıyasla öne çıkmıştır. PatchTST ise benzer şekilde yüksek bir doğruluk (%53) ve F1 skoru (0.49) ile Autoformer'ın hemen ardından gelmiştir. Bu iki modelin görece daha iyi performans sergilemesinde, zaman serisindeki kısa ve orta vadeli örüntüleri yakalamadaki etkinliklerinin etkili olduğu düşünülmektedir. Diğer yandan, Informer ve Reformer modelleri beklentilerin altında kalmış, özellikle Reformer %28 gibi düşük bir doğruluk oranıyla sınıflar arasında yeterli ayrımı gerçekleştirememiştir. TimesNet modeli ise doğruluk ve F1 skoru açısından orta seviyede kalmış, ancak Autoformer ve PatchTST kadar istikrarlı bir başarı gösterememiştir.

Elde edilen bu sonuçlar, kullanılan modellerin performanslarını net bir şekilde ortaya koysa da, bu başarımların düzeylerinin yalnızca model mimarisiyle sınırlı olmadığı göz önünde bulundurulmalıdır. Futbol, doğası gereği birçok değişkene bağlı, karmaşık ve belirsizlik barındıran bir oyun yapısına sahiptir. Oyuncu sakatlıkları, kadro rotasyonları, hakem kararları gibi modelle dışsallaştırılması zor etkenler, tahmin performansını doğal olarak sınırlandırmaktadır. Bunun yanında, elimizdeki veri seti niceliksel ve niteliksel anlamda daha zengin olsaydı, modellerin öğrenme kapasitesi artabilir ve sonuçlar daha da iyileştirilebilirdi. Özellikle bazı takımların geçmiş verilerine ulaşamama, belirli istatistiklerin eksik ya da dengesiz olması gibi sınırlılıklar, tahmin doğruluğunu olumsuz yönde etkilemiş olabilir. Tüm bu değerlendirmeler sonucunda, zaman serisi temelli mimarilerin futbol verisi üzerinde belirli bir başarı elde edebildiği görülmüştür. Ancak gerçek dünya uygulamaları için daha büyük, dengeli ve güncel veri setlerine ihtiyaç duyulduğu açıktır.

#### KAYNAKÇA

- [1] Taşpınar, Y. S., Çınar, İ., & Koklu, M. (2021). Improvement of Football Match Score Prediction by Selecting Effective Features for Italy Serie A League. *MANAS Journal of Engineering*, 9(1), 1-9. <https://doi.org/10.51354/mjen.802818>
- [2] Zaveri, Nilay, et al. "Prediction of football match score and decision making process." *International Journal on Recent and Innovation Trends in Computing and Communication* 6.2 (2018): 162-165.
- [3] Igiri, Chinwe Peace, and Enoch Okechukwu Nwachukwu. "An improved prediction system for football a match result." *IOSR journal of Engineering* 4.12 (2014): 12-20.
- [4] <https://arsiv.mackolik.com/Teams/OptaList.aspx>
- [5] <https://medium.com/machine-learning-t%C3%BCrkiye/korelasyon-nedir-python-ile-korelasyon-analizi-fbd24deb20dd>
- [6] <https://gokerguner.medium.com/machine-learning-2-korelasyon-matrisi-%C3%B6zellik-se%C3%A7imi-s%C4%B1n%C4%B1flar%C4%B1n-dengesizli%C4%9Fi-karar-a%C4%9Fa%C3%A7lar%C4%B1-af993bd8ea66>

- [7] <https://medium.com/@halimedogan/makine-%C3%B6%C4%9Frenimi-veri-k%C3%BCmenizdeki-dengesiz-s%C4%B1n%C4%B1flar-m%C3%BCcadele-i%C3%A7in-taktikler-b99290ee8c88>
- [8] <https://www.btkakademi.gov.tr/portal/course/veri-bilimi-icin-python-ve-tensorflow-11705>
- [9] <https://github.com/ctxj/Time-Series-Transformer-Pytorch>
- [10] <https://www.kaggle.com/code/shreyasajal/pytorch-forecasting-for-time-series-forecasting>

- [Drive Veri Toplama](#)
- [Drive Model Link I](#)
- [Drive Model Link II](#)
- [GitHub Veri Toplama Link 1](#)
- [GitHub Modelleme Link 2](#)

