

Petrol Fiyatları Gelecek Öngörülleri: Yapay Zeka Destekli Tahmin Modelleri

Nilay Başol
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
231307096@kocaeli.edu.tr

Yusuf Perktas
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
231307103@kocaeli.edu.tr

Özet— Bu çalışma, petrol fiyatlarının tahmin edilmesi için farklı transformatör modellerinin kullanılarak karşılaştırmasını içermektedir. Projenin ilk adımında toplanan veriler, farklı transformatör mimarileri (Autoformer, Informer, Reformer, Vanilla Transformer ve ETT) kullanılarak analiz edilmiş ve elde edilen sonuçlar değerlendirilmiştir. Proje, veri toplama, temizleme, modelleme ve sonuç analizi gibi aşamaları kapsamaktadır. Çalışma sonuçları, farklı transformatör modellerinin petrol fiyat tahmini performanslarını göstermektedir.

Anahtar Kelimeler— Transformatör Modelleri, Zaman Serisi Analizi, Petrol Fiyat Tahmini, Yapay Zeka, Autoformer, Informer, Reformer, Vanilla Transformer, ETT

I. GİRİŞ

Petrol, dünyadaki ekonomik ve politik dengeleri şekillendiren en önemli enerji kaynaklarından biridir. Endüstriyel faaliyetlerden ulaşma kadar geniş bir yelpazede kullanılması, petrolün fiyat hareketlerinin büyük bir ilgiyle takip edilmesine neden olmaktadır. Fiyatlar, üretim kapasitesindeki değişiklikler, enerji politikalarındaki reformlar ve teknolojik gelişmeler gibi birçok değişkenin etkisi altındadır. Özellikle enerji sektöründeki jeopolitik riskler, petrol fiyatlarında öngörülemeyen dalgalanmalara yol açabilmektedir. Bu nedenle, petrol fiyatlarının doğru tahmin edilmesi, enerji politikaları ve finansal stratejilerin belirlenmesinde kritik bir rol oynamaktadır.

Geleneksel tahmin yöntemleri, zaman serisi analizleri ve istatistiksel modellerle sınırlı kalmaktadır. Ancak yapay zeka ve makine öğrenmesi tekniklerinin gelişmesiyle, daha karmaşık ve dinamik modeller bu alanda başarıyla uygulanmaya başlanmıştır. Yapay zeka modelleri, büyük veri setlerinden elde edilen bilgileri analiz ederek karmaşık ve doğrusal olmayan ilişkileri daha etkili bir şekilde ortaya çıkarabilir. Bu sayede, geleneksel yöntemlerin yetersiz kaldığı dalgalanmalar ve ani fiyat değişiklikleri daha önceden tespit edilebilmektedir.

Transformatör modelleri, çok boyutlu ve büyük verileri etkili bir şekilde işleyerek uzun vadeli bağımlılıkları öğrenmede öne çıkan yapay zeka modelleridir. Zaman serisi verileri üzerinde oldukça başarılı sonuçlar veren bu modeller,

finansal tahminleme, enerji analizi ve iklim verileri gibi çeşitli alanlarda kullanılmaktadır.

Bu çalışma, farklı transformatör modellerinin petrol fiyatları tahminindeki performanslarını geniş bir perspektiften değerlendirmekte ve modellerin karmaşık veri yapıları üzerinde nasıl performans gösterdiğini detaylı bir şekilde analiz etmektedir. Hangi modelin farklı veri setlerinde daha düşük hata oranları verdiği ve istikrarlı sonuçlar sağladığı belirlenerek, uygulamalı senaryolar için en uygun transformatör mimarisi ortaya konmaktadır.

II. YÖNTEM

Bu bölümde, kullanılan veri seti ve modellerin eğitimi hakkında detaylı bilgiler verilecektir.

A. Veri Seti

1) Veri Toplama Süreci:

Veri toplama işlemi, Python programlama dili ve Selenium kütüphanesi kullanılarak yapılmıştır. Selenium, web sayfalarından veri çekmek için güçlü bir araçtır ve dinamik olarak yüklenen içerikleri de işleyebilme kapasitesine sahiptir. Bu proje için, Selenium ile Yahoo Finance web sitesindeki tablolardan veri çekilmiştir. Toplanan veriler, CSV formatında saklanmıştır. Bu format, verilerin rahatlıkla işlenebilmesi ve modelleme için kullanılabilmesi adına uygun bir yöntemdir. Her bir satırda tarih ve fiyat bilgileriyle birlikte, ham petrol fiyatları ve indikatörlere bağlı verilerin her biri belirli bir formatta saklanmıştır.

2) Veri Setinin Yüklenmesi:

Bu çalışmada kullanılan veri seti, Google Drive'daki "Transformer-main" klasöründe bulunan "crude_oil_data_with_indicators.csv" dosyasından alınmaktadır. Veri seti, saatlik olarak toplanmış petrol fiyatlarına ve çeşitli ekonomik göstergelere ilişkin verilerdir. Veriler, özellikle petrol fiyatlarının gelecekteki hareketlerini tahmin etmek amacıyla kullanılmıştır.

Veri setinde yer alan başlıca sütunlar şunlardır:

- **Açılış (Open), Kapanış (Close), En Yüksek (High), En Düşük (Low):** Petrolün bir işlem günündeki fiyat hareketlerini temsil eder.
- **MACD (Moving Average Convergence Divergence), MACD_Signal, RSI (Relative Strength Index), Upper_Band, Middle_Band, Lower_Band:** Teknik analizde sıklıkla kullanılan göstergelerdir. Bu göstergeler, piyasa trendlerinin ve olası dönüş noktalarının belirlenmesinde yardımcı olur.
- **EMA (Exponential Moving Average):** 25, 50, 100 ve 200 periyotluk EMA değerleri, fiyat hareketlerinin belirli bir süre içindeki ortalama değerini belirleyerek, piyasa eğilimlerini gösterir.

Bu veri seti, zaman serisi analizi ve modelleme için zengin bir kaynak sunmakta olup, petrol fiyatları ile çeşitli teknik analiz göstergelerini içermektedir. Bu sayede modelin daha doğru tahminler yapabilmesi amaçlanmıştır.

3) Eksik Veri Analizi:

Veri seti üzerinde yapılan analizler sonucunda, herhangi bir eksik veriye rastlanmamıştır. Bu durum, veri temizleme aşamasını önemli ölçüde basitleştirerek süreci hızlandırmıştır. Eksik verilerin olmaması, modelleme sürecine geçmeden önce herhangi bir veri doldurma veya çıkarma işlemine ihtiyaç duyulmadığı anlamına gelmektedir. Bu nedenle, veri seti üzerinde eksik veriyle ilgili herhangi bir ön işleme adımı uygulanmamıştır. Bu durum, modelin doğruluğunu etkileyecek herhangi bir veri kaybı veya yanlışlık riski yaratmadığından, analiz sürecinin sağlıklı bir şekilde devam etmesine olanak sağlamıştır.

4) Veri Tipi Dönüşümleri:

Veri setindeki mevcut sütunlar incelendiğinde, veri tipi dönüşümüne ihtiyaç duyulmadığı görülmüştür. Zaman verisi uygun formatta sunulmuş olup, sayısal sütunlar da doğru veri tiplerinde yer almaktadır. Bu nedenle, veri tipi dönüşüm işlemleri yapılmamıştır.

5) Ön İşlem Öncesi Veri Görselleştirmesi:

Veri görselleştirme, veri setinin yapısını anlamak ve modelleme öncesi gerekli işlemleri belirlemek için önemli bir adımdır. Bu aşamada, veri setindeki önemli sütunların dağılımları ve ilişkileri incelenmiştir.

A. Sayısal Sütunların Histogramları:

Bu adımda, veri setindeki sayısal sütunların dağılımlarını anlamak, sütunlar arasındaki ilişkileri belirlemek ve eksik değerleri kontrol etmek amacıyla sayısal veri analizi yapılmıştır. Elde edilen analiz sonuçları, aşağıdaki görselde detaylı bir şekilde sunulmuştur.

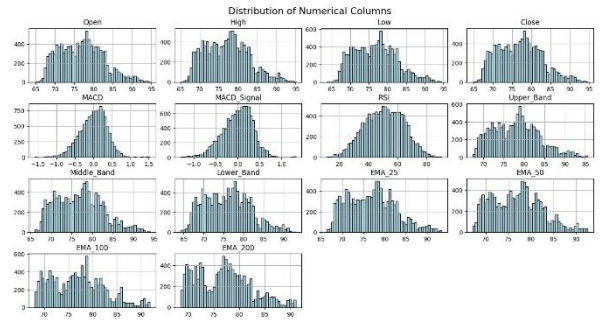


Fig. 1: Veri setindeki sayısal sütunların dağılımlarını gösteren histogramlar.

Histogramlar, her bir sütunun veri setindeki değerlerinin frekansını ve dağılımını görsel olarak göstererek, verinin genel yapısını anlamamıza yardımcı olmaktadır. Bu görselleştirme, verinin normal dağılıma uygunluğunu, olası aykırı değerleri ve veri setinin genel özelliklerini gözlemlememizi sağlamaktadır. Özellikle **açılış, kapanış, yüksek ve düşük** fiyatlar gibi temel sütunların dağılımları, fiyat hareketlerinin zaman içindeki dalgalanmasını ve piyasa koşullarını yansıtmak için önemlidir. Ayrıca, teknik analiz göstergeleri olan **MACD, RSI, Upper_Band, Middle_Band, Lower_Band** gibi sütunlar, piyasa eğilimlerini ve olası dönüş noktalarını belirlemek adına değerli bilgiler sunmaktadır.

Görselleştirme sürecinde elde edilen histogramlar, sütunlar arasındaki ilişkiler ve verinin potansiyel problemleri hakkında fikir edinmemizi sağlar. Örneğin, bazı sütunların normal dağılıma yakın olduğu, bazılarının ise belirgin bir şekilde çarpık veya farklı bir dağılıma sahip olduğu gözlemlenebilir. Bu tür analizler, modelleme sürecinde hangi özelliklerin daha etkili olacağı ve hangi ön işleme adımlarının gerekli olacağı konusunda bize rehberlik etmektedir.

B. Genel Korelasyon Yapısı:

Veri setindeki değişkenler arasındaki ilişkiyi anlamak için yapılan korelasyon analizi, değişkenler arasında karmaşık bir ilişki ağının varlığını ortaya koymuştur. Korelasyon matrisi, her bir sütunun diğer sütunlarla olan doğrusal ilişkisini gösteren bir araçtır. Normal veri seti üzerinde yapılan analizde, bazı değişkenler arasında güçlü pozitif veya negatif korelasyonlar gözlemlenmiştir.

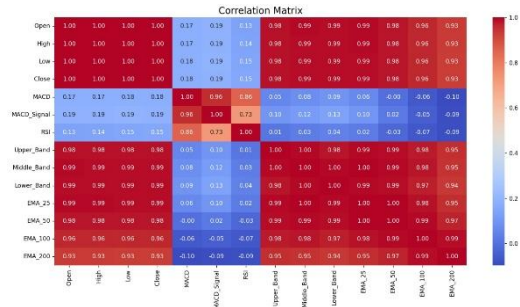


Fig. 2: Veri setindeki sayısal değişkenler arasındaki korelasyon matrisini göstermektedir. Matristeki değerler, her bir değişkenin diğer değişkenlerle olan doğrusal ilişkisini temsil etmektedir. Yüksek pozitif korelasyonlar, değişkenlerin birbirine paralel hareket ettiğini, negatif korelasyonlar ise zıt yönde hareket ettiklerini göstermektedir.

Bu, belirli sütunların birbiriyle yakın ilişkili olduğunu ve birindeki değişimin diğerini etkileyebileceğini göstermektedir. Örneğin, 'Open' (Açılış) ve 'Close' (Kapanış) fiyatları arasında yüksek pozitif korelasyon, bu iki sütunun birbirine paralel hareket ettiğini ve genellikle birbirleriyle uyumlu olduğunu gösteriyor olabilir. Diğer yandan, 'MACD' ve 'RSI' gibi teknik analiz göstergelerinin korelasyonu, piyasa eğilimlerinin farklı açılardan analiz edilmesi gerektiğini ortaya koymaktadır.

Korelasyon yapısı, veri setindeki hangi değişkenlerin birlikte hareket ettiğini ve hangilerinin birbirinden bağımsız olduğunu anlamamıza yardımcı olur. Bu tür bir analiz, modelin doğruluğunu artırmak için hangi özelliklerin daha anlamlı olacağını belirlemede önemlidir. Ayrıca, bazı sütunlar arasındaki yüksek korelasyonlar, multikolinearite gibi problemlere yol açabileceğinden, bu durum modelleme aşamasında dikkatle değerlendirilmelidir.

B. Modeller

Bu bölümde, petrol fiyat tahmininde kullanılan beş farklı modelin detaylı açıklamaları yapılacaktır. Her modelin temel yapısı, çalışma prensipleri, eğitim süreçleri ve zaman dilimlerine göre modellerin performansları üzerine odaklanılacaktır.

1. Model Yapıları ve Özellikleri:

a. Autoformer:

Autoformer, zaman serisi tahminlerinde etkili sonuçlar veren modern bir modeldir ve özellikle zaman serilerindeki trend (uzun vadeli değişim) ve mevsimsellik (dönemsel değişiklikler) gibi bileşenleri öğrenmek için tasarlanmıştır. Autoformer, klasik Transformer yapısını zaman serisi verilerine adapte eden bir modeldir. Modelin en önemli özelliklerinden biri, zaman serisindeki sezonsal ve trend bileşenlerini ayrı ayrı işlemesi ve bu bileşenlerin her birini daha hassas bir şekilde öğrenmesidir.

- **Trend ve Sezonluk Bileşenler:** Autoformer, zaman serisi verisini iki ana bileşene ayırarak işler: trend bileşeni ve sezonluk bileşeni. Bu bileşenlerin her biri, modelin kendi alt katmanlarında işlenir. Bu özellik, zaman serisindeki hem uzun vadeli eğilimleri (trend) hem de kısa vadeli döngüleri (sezonluk etkiler) modellemede büyük bir avantaj sağlar.
- **Transformer Yapısı:** Autoformer, klasik Transformer modelinin encoder-decoder yapısını kullanır. Ancak, zaman serisi verilerine uyum sağlamak amacıyla, modeldeki dikkat mekanizmaları (attention mechanisms) ve katmanlar, verinin zaman bağımlılıklarını daha verimli öğrenmek üzere optimize edilmiştir.

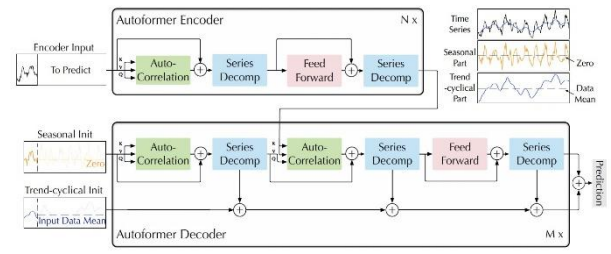


Fig. 3: Autoformer'in Genel Mimarisi

Transformer tabanlı yapısında yer alan dikkat mekanizmaları, Autoformer'a geçmiş verilerden geleceğe doğru tahminler yapma yeteneği kazandırır. Bu mekanizma, verinin sıralı yapısını dikkate alarak her bir zaman adımıdaki önemli bilgiyi ön plana çıkarır.

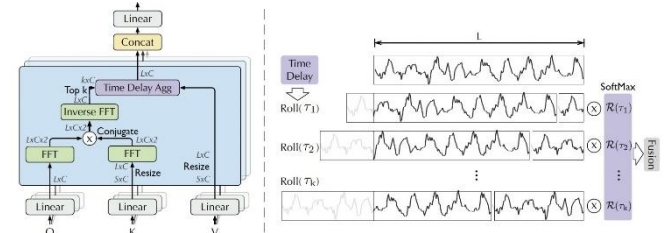


Fig. 4: Auto-Correlation mekanizması

b. Informer:

Informer, özellikle büyük ölçekli zaman serisi verileri için geliştirilmiş bir Transformer modelidir. Zaman serisi tahminlerine yönelik yapılan geleneksel modelleme yöntemlerinden farklı olarak, Informer, verideki uzun vadeli bağımlılıkları daha verimli bir şekilde öğrenmek ve büyük veri setleriyle başa çıkabilmek amacıyla tasarlanmıştır. Modelin en belirgin özelliklerinden biri, sparse attention mechanism (seyrek dikkat mekanizması) kullanarak daha hızlı ve daha verimli sonuçlar elde etmesidir.

Informer, klasik Transformer yapısına dayansa da zaman serisi verilerine özel geliştirilmiş birkaç bileşene sahiptir. Bu bileşenler, büyük veri setlerinde daha hızlı eğitim süreci ve daha düşük hesaplama maliyetleri ile yüksek doğruluk sağlamak için optimize edilmiştir.

- **Seyrek Dikkat Mekanizması (Sparse Attention):** Transformer'ın dikkat mekanizmaları, her bir zaman dilimindeki tüm veriyi dikkate alır, bu da özellikle büyük veri setlerinde yüksek hesaplama maliyetlerine yol açar. Informer, bu sorunu çözmek için sparse attention (seyrek dikkat) tekniği kullanır. Bu teknik, yalnızca önemli zaman dilimlerini dikkate alarak modelin hesaplama yükünü önemli ölçüde azaltır ve verinin öğrenilmesinde hız sağlar.
- **Probabilistic Selection of Top-K Keys:** Informer, dikkat mekanizmasını iyileştirmek için probabilistic selection (olasılıksal seçim) kullanır. Bu sayede, her bir zaman diliminde önemli olan veriler belirlenerek, modelin daha verimli öğrenmesi sağlanır.
- **Encoder-Decoder Yapısı:** Informer, klasik Transformer modelinde olduğu gibi encoder-decoder yapısını kullanır. Encoder kısmı geçmiş verilerden bilgiyi alırken, decoder kısmı bu

bilgileri kullanarak geleceğe yönelik tahminler yapar. Bu yapı, modelin sıralı verilerdeki bağımlılıkları anlamasında güçlüdür.

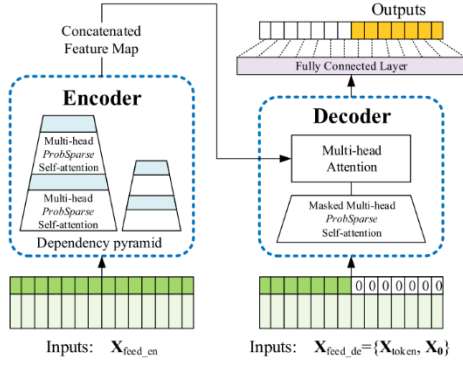


Fig. 5: Informer'in Mimarisi

c. Reformer:

Reformer, Transformer modeli, özellikle büyük veri setlerinde eğitim yaparken karşılaşılan hesaplama ve bellek gereksinimlerini azaltmayı hedefler. Reformer, geleneksel Transformer'ın güçlü yanlarını korurken, daha verimli bir şekilde çalışacak şekilde optimize edilmiştir. Bu model, çok büyük veri setlerinde bile daha hızlı eğitim ve daha düşük bellek kullanımı ile güçlü tahminler yapabilir.

Reformer'ın en dikkat çeken yeniliklerinden biri, sıkıştırılmış dikkat mekanizması (locality-sensitive hashing, LSH) ve dönüşümlü (reversible) katmanlar kullanmasıdır. Bu yenilikçi yöntemler, büyük veri setlerinin öğrenilmesi sürecinde hesaplama maliyetini büyük ölçüde azaltır.

- **Sıkıştırılmış Dikkat Mekanizması (Locality-Sensitive Hashing - LSH):** Transformer'da kullanılan dikkat mekanizması, her bir giriş için tüm diğer girişlere olan bağımlılıkları hesaplamak zorundadır. Bu, özellikle büyük veri setlerinde büyük bir hesaplama yükü oluşturur. Reformer, locality-sensitive hashing (LSH) tekniğini kullanarak bu bağımlılıkları yalnızca benzer girişler arasında hesaplamaya indirger. Bu sayede, her dikkat işlemi daha verimli hale gelir ve hesaplama maliyeti büyük ölçüde azalır.

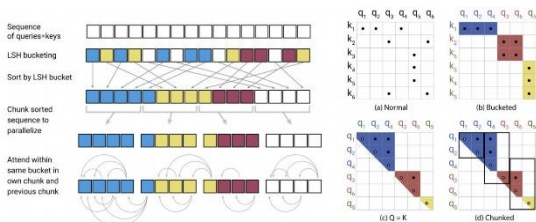


Fig. 6: Reformer Modelindeki Dikkat Mekanizması

- **Reversible Layers (Dönüşümlü Katmanlar):** Geleneksel Transformer modelindeki her bir katman, giriş ve çıkış verilerini saklamak zorundadır. Bu, belleğin hızlı bir şekilde tükenmesine yol açabilir. Reformer, dönüşümlü katmanlar kullanarak, katmanların çıktıları yerine sadece girişleri saklar. Bu yöntem, bellek kullanımını büyük ölçüde azaltarak modelin daha büyük veri setleri üzerinde çalışmasını sağlar.

- **Attention Mechanism Optimization:** Reformer, Transformer'daki klasik dikkat mekanizmasını optimize eder. Seyrek dikkat (sparse attention) ve LSH gibi tekniklerle, dikkat işlemleri daha verimli hale gelir ve modelin her bir zaman dilimindeki önemli bilgileri daha hızlı bir şekilde öğrenmesi sağlanır.

d. Vanilla Transformer:

Vanilla Transformer, temel Transformer modelinin ilk sürümüdür ve zaman serisi tahmininden doğal dil işlemeye kadar birçok alanda güçlü performans göstermektedir.

2017 yılında Vaswani ve arkadaşları tarafından önerilen bu model, self-attention mekanizması kullanarak, sıralı verilerdeki ilişkileri öğrenmede çok başarılıdır.

Vanilla Transformer, encoder-decoder yapısına sahip bir modeldir. Her iki yapı da sıralı verilerdeki bağımlılıkları öğrenebilmek için bir dizi dikkat katmanına sahiptir.

- **Self-Attention Mekanizması:** Self-attention, her bir zaman dilimindeki verinin, diğer zaman dilimlerine olan ilişkisini öğrenir. Bu mekanizma, özellikle uzun vadeli bağımlılıkları öğrenmekte etkilidir. Girdi verisindeki her bir öge, diğer ögelere olan "dikkatini" hesaplayarak, önemli olan bilgiyi çıkarır.
- **Multi-Head Attention:** Self-attention'ın farklı bölümlerini paralel olarak işlemek için kullanılan multi-head attention, verinin farklı yönlerine dair bilgi toplar. Bu sayede, model daha zengin temsiller öğrenebilir.
- **Positional Encoding:** Zaman serisi verilerinde sıralı ilişkiler çok önemlidir. Positional encoding, giriş verisine zaman bilgisini ekleyerek, modelin sıralı verilerdeki bağımlılıkları doğru bir şekilde öğrenmesini sağlar.
- **Feedforward Layers ve Normalizasyon:** Her dikkat katmanının ardından, veriler bir feedforward ağından geçirilir ve sonuçlar normalizasyon katmanlarıyla iyileştirilir. Bu işlemler, modelin daha hızlı ve verimli çalışmasına yardımcı olur.

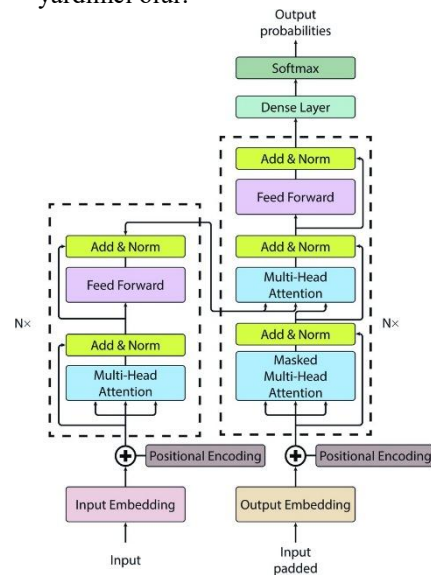


Fig. 7: Vanilla Transformer Mimarisi

e. ETT :

Exponential Transformer (ETT), zaman serisi verilerinde kullanılan bir modeldir ve özellikle zaman serisi tahminlerinde performansı artırmak amacıyla geliştirilmiştir. ETT, geleneksel Transformer'ın kapasitesine ek olarak zamanla değişen veri dinamiklerini modelleyebilecek şekilde tasarlanmıştır.

ETT, Transformer tabanlı modelin temel yapısını kullanırken, zaman serisi verilerine özgü yenilikler ekler. Bu yenilikler, özellikle verinin daha doğru bir şekilde öğrenilmesine olanak tanır.

- **Exponential Smoothing (Üssel Düzgünleştirme):** ETT, zaman serisi verilerindeki trendleri daha iyi öğrenebilmek için üssel düzgünleştirme (exponential smoothing) tekniklerinden yararlanır. Bu yöntem, modelin daha az gürültüyle doğru trendleri öğrenmesini sağlar.
- **Exponential Time Attention:** Zaman serisi verilerindeki dinamikleri daha verimli şekilde öğrenebilmek için, ETT, geleneksel dikkat mekanizmasından farklı olarak **zaman odaklı dikkat** (time attention) kullanır. Bu, verinin zaman içinde değişen özelliklerini daha iyi yakalayabilmesini sağlar.
- **Dynamic Time Windowing:** ETT, veriye dinamik zaman pencereleriyle yaklaşır. Bu, modelin kısa vadeli ve uzun vadeli dinamikleri daha doğru bir şekilde öğrenmesine yardımcı olur. Dinamik zaman penceresi, sadece önemli olan zaman dilimlerine dikkat edilmesini sağlar.

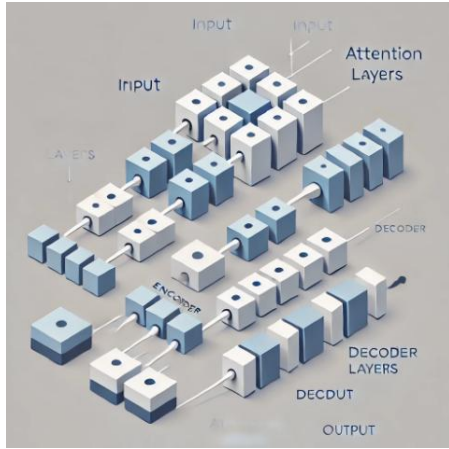


Fig. 8: ETT Genel Mimarisi

2. Model Eğitim ve Test Süreçleri

Bu bölümde, projede kullanılan modellerin (Autoformer, Informer, Reformer, Vanilla Transformer ve ETT) eğitim süreçleri ve projedeki kullanım amaçları detaylı bir şekilde açıklanmaktadır. Modellerin her biri, zaman serisi verileri üzerinde farklı perspektifler sunarak petrol fiyat tahminlerinde daha doğru ve verimli sonuçlar elde edilmesini hedeflemektedir.

a) Autoformer:

Autoformer modeli, zaman serisi verilerinin doğasında var olan sezonsal ve trend bileşenlerini ayrıştırarak çalışır. Eğitim sürecinde model, girdi

verilerini sezonsal ve trend bileşenlerine ayırır, bu bileşenleri ayrı ayrı işler ve ardından tekrar birleştirilerek tahminler yapar. Bu yöntem, modelin verideki kalıpları daha verimli bir şekilde öğrenmesini sağlar.

Kodun başlangıcında, modelin doğru ve tutarlı sonuçlar verebilmesi için random seed ayarlaması yapılmaktadır. Bu, sonuçların tekrarlanabilir olmasını sağlar. Ardından, eğitim sürecinde kullanılacak olan parametreler argparse ile tanımlanır. Bu parametreler modelin detaylı ayarlarını, veri seti yolunu ve eğitim sürecine dair temel yapılandırmaları içerir. Modelin girdi olarak aldığı veri dizisinin uzunluğu 96 saatlik veri, girdi dizisinde kullanılan etiket verisi uzunluğu 48 saat, modelin tahmin ettiği veri uzunluğu 72 saattir. Modelin eğitim süreci, Exp_Main sınıfı üzerinden başlatılmaktadır. Kodun önemli bir kısmı, modeli belirlenen sayıda tekrar (iteration) boyunca eğitmektedir.

Eğitim tamamlandıktan sonra, model test verisi üzerinde değerlendirilir. Test süreci, eğitimle aynı yapı kullanılarak gerçekleştirilir. Bu süreçte, modelin doğruluğu ve performansı ölçülerek kaydedilir. Test süreci boyunca GPU belleği temizlenir.

Aşağıda modelin eğitim sürecinde elde edilen epoch bazında kayıp (loss) grafiği sunulmaktadır. Bu grafik, modelin performansını ve eğitim sürecindeki iyileşmeleri görsel olarak değerlendirmeye yardımcı olacaktır.

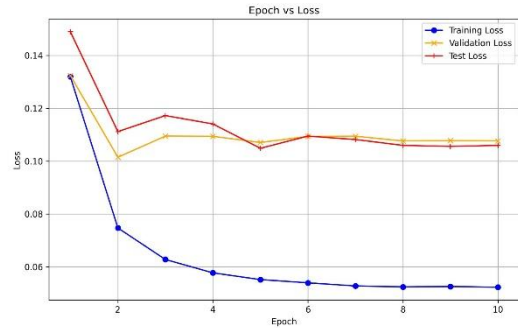


Fig. 9: Autoformer Epoch Loss Grafiği

b) Informer:

Informer modelinin eğitim sürecinde belirli parametreler kullanılmaktadır. Kodun başlangıcında, modelin çalışma ortamı ayarlanır ve sabit bir seed belirlenerek modelin tekrarlanabilirliği sağlanır. Bu, random, torch ve numpy kütüphaneleri üzerinden yapılmaktadır.

Ardından, argparse yapısı kullanılarak modelin temel parametreleri belirlenir. Bu parametreler, modelin türü, veri yolu, tahmin süresi, veri özellikleri ve eğitim süresi gibi faktörleri içermektedir.

Modelin girdi olarak aldığı veri dizisinin uzunluğu 96 saatlik veri, girdi dizisinde kullanılan etiket verisi uzunluğu 48 saat, modelin tahmin ettiği veri uzunluğu 72 saattir.

Modelin eğitimi, belirlenen parametreler doğrultusunda yürütülmektedir. Eğitim işlemi için, deney ortamı (Exp_Main) tanımlanır ve modelin ayarları bu ortam üzerinden yürütülür. Eğitim süreci tamamlandıktan sonra model test edilir. Test işlemi, aynı ayarlar kullanılarak modelin performansını değerlendirir. Eğitim ve test süreçlerinde GPU kullanımı sağlanarak işlemler hızlandırılmaktadır.

Aşağıda modelin eğitim sürecinde elde edilen epoch bazında kayıp (loss) grafiği sunulmaktadır. Bu grafik, modelin performansını ve eğitim sürecindeki iyileşmeleri görsel olarak değerlendirmeye yardımcı olacaktır.

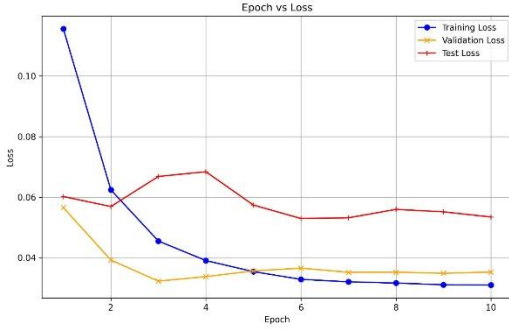


Fig. 10: Informer Epoch Loss Grafiği

c) Reformer:

Modelin eğitim sürecinde belirli parametreler kullanılmaktadır. İlk olarak, argparse kütüphanesi kullanılarak deney parametreleri ayarlanır. Bu parametreler model konfigürasyonu, veri parametreleri ve eğitim ayarlarını içerir. Yeniden üretilebilirlik için NumPy, Python'un rastgele kütüphanesi ve PyTorch için sabit bir rastgele seed ayarlanır. Bu, eğitim sürecinin birden fazla kez çalıştırıldığında tutarlı sonuçlar üretmesini sağlar.

Reformer modeli, Exp_Main sınıfından başlatılır ve deney parametreleri geçirilir. Bu sınıf, modeli kurmak, veriyi yüklemek, model mimarisini tanımlamak ve eğitim ve test süreçlerini yönetmek için sorumludur.

Eğitim süreci, deneyin birden fazla yinelenmesi için bir döngü içinde başlatılır (for ii in range(args.itr)). Model, train() metodu ile eğitilir ve kayıp fonksiyonu (Bu projede, Ortalamadan Kare Hatası - mse) kullanılarak ağırlıklar ayarlanır. Eğitim aşamasının çalışma süresi ölçülür ve her deney için süre kaydedilir.

Eğitim sonrası model, test() metodu ile test edilir ve daha önce görmediği test verileri üzerinde değerlendirilir. Bu, modelin yeni verilere nasıl genelleme yaptığına dair fikir verir.

Kod, hızlandırılmış eğitim için GPU kullanımı (torch.cuda) etkinleştirir ve her deneyden sonra GPU belleğini temizler, böylece sonraki çalışmalar için bellek sorunlarının önüne geçilir.

Aşağıda modelin eğitim sürecinde elde edilen epoch bazında kayıp (loss) grafiği sunulmaktadır.

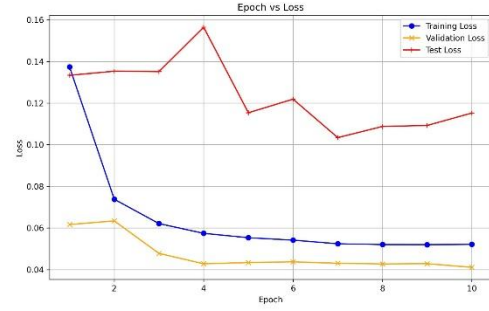


Fig. 11: Reformer Epoch Loss Grafiği

d) Vanilla Transformer:

Vanilla Transformer modelinin eğitim sürecinde belirli parametreler kullanılmaktadır. Eğitim sürecinde sabırlı ve tekrarlanabilir sonuçlar almak için rastgele seed ayarlanır. Bu, eğitimdeki herhangi bir rastgele bileşenin aynı başlangıç değerlerine sahip olmasını sağlar.

Modelin girdi olarak aldığı veri dizisinin uzunluğu 96 saatlik veri, girdi dizisinde kullanılan etiket verisi uzunluğu 48 saat, modelin tahmin ettiği veri uzunluğu 72 saattir.

Model, her veri örneği için çıkış değerini tahmin etmek üzere eğitim verisini (girdi) geçirir. Bu aşama, zaman serisinin önceki verilerine bakarak (yani giriş verileri) gelecekteki değerleri tahmin etmek için Transformer mimarisinin dikkat (attention) mekanizmasını kullanır.

Loss fonksiyonu olarak MSE (Mean Squared Error) kullanılır. Modelin tahminleri ile gerçek değerler arasındaki farkın karesinin ortalamasıdır. Bu, modelin doğruluğunu ölçen temel bir metriktir. Kayıp fonksiyonu, modelin eğitilmesi sırasında minimize edilmesi gereken bir değerdir.

Eğitim tamamlandıktan sonra model test edilir. Bu aşama, modelin daha önce görmediği veri üzerinde performansını ölçmek amacıyla yapılır.

Kod, hızlandırılmış eğitim için GPU kullanımı (torch.cuda) etkinleştirir ve her deneyden sonra GPU belleğini temizler, böylece sonraki çalışmalar için bellek sorunlarının önüne geçilir.

Aşağıda modelin eğitim sürecinde elde edilen epoch bazında kayıp (loss) grafiği sunulmaktadır.

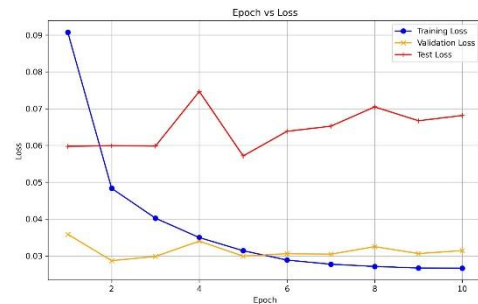


Fig. 12: Vanilla Transformer Loss Grafiği

e) ETT:

ETT modelinin eğitim süreci, zaman serisi verilerini tahmin etmek için güçlü bir yöntem sunar. İlk adımda, veriler MinMaxScaler kullanılarak ölçeklendirilir, böylece modelin öğrenmesi daha hızlı ve verimli hale gelir. Veriler eğitim ve test setlerine ayrıldıktan sonra, geçmiş verilerden gelecekteki değerleri tahmin etmek için bir zaman penceresi (time_step) kullanılır. Modelin yapısı, Transformer Encoder katmanları ve tam bağlantılı (fully connected) katmanlardan oluşur.

Eğitim süreci, modelin tahminleriyle gerçek değerler arasındaki farkı minimize etmeyi amaçlayan bir kayıp fonksiyonu (MSE) ile gerçekleştirilir. Her epoch'ta eğitim kayıpları hesaplanır ve modelin performansı test verileriyle değerlendirilir. Eğitim süresi de kaydedilerek modelin verimliliği ölçülür. Son olarak, eğitim ve test kayıplarının grafiksel olarak görselleştirilmesi, modelin öğrenme sürecini anlamak için faydalıdır. Bu süreç, zaman serisi tahminlerinin doğruluğunu artırarak daha güvenilir sonuçlar elde edilmesine olanak sağlar.

Aşağıda modelin eğitim sürecinde elde edilen epoch bazında kayıp (loss) grafiği sunulmaktadır.

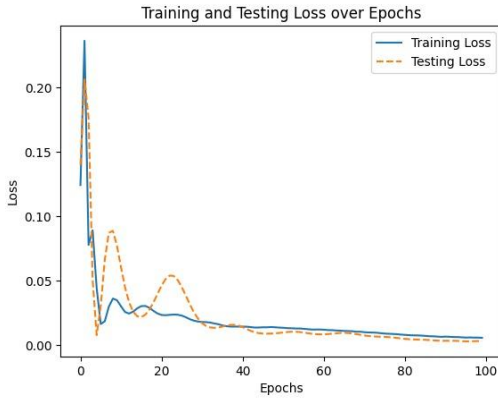


Fig. 13: ETT Loss Grafiği

3. Modellerin Performans Metrikleri

Aşağıda, modellerin farklı zaman dilimleri (96, 72, 48, 24) için performans metriklerini karşılaştıran tablolar yer almaktadır. Bu tablolar, her bir modelin tahmin doğruluğunu ve çeşitli metriklerle dayalı başarısını daha iyi anlamamıza olanak sağlayacaktır.

Tablolar, model karşılaştırmalarını kolaylaştırmak için kullanılan Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) ve R-Squared gibi önemli metrikler doğrultusunda düzenlenmiştir. Bu metrikler, modellerin zaman dilimlerine göre nasıl performans gösterdiğini karşılaştırmamıza yardımcı olacaktır.

Metric	Length	Autoformer	Reformer	Informer	Vanilla
MSE	96	0.247	0.437	0.233	0.145
MAE	96	0.394	0.574	0.402	0.293
RMSE	96	0.497	0.661	0.482	0.38
MAPE	96	1.127	1.097	1.032	1.048
MSPE	96	295.359	157.362	234.146	228.118
R2	96	0.205	-0.41	0.25	0.534

Fig. 14: 96 Length Metric

Bu tablo, 96 saatlik zaman diliminde tahminler yapan modellerin performansını göstermektedir. Bu uzun zaman diliminde, modellerin daha fazla belirsizlikle karşı karşıya kalması ve dolayısıyla daha yüksek hata oranları üretmesi beklenebilir. Bu durum, uzun vadeli tahminlerin zorluklarını yansıtmaktadır.

Metric	Length	Autoformer	Reformer	Informer	Vanilla
MSE	72	0.228	0.251	0.191	0.157
MAE	72	0.373	0.421	0.356	0.317
RMSE	72	0.477	0.501	0.437	0.396
MAPE	72	1.156	1.042	0.955	0.898
MSPE	72	172.438	215.633	232.806	180.353
R2	72	0.272	0.198	0.389	0.499

Fig. 15: 72 Length Metric

72 saatlik zaman diliminde modellerin performansı, 96 saatlik zaman dilimine göre daha iyidir, çünkü kısa vadeli tahminler, daha fazla veriye dayalı ve daha az belirsizlik içerir.

Metric	Length	Autoformer	Reformer	Informer	Vanilla
MSE	48	0.196	0.198	0.118	0.124
MAE	48	0.338	0.374	0.279	0.279
RMSE	48	0.443	0.445	0.343	0.353
MAPE	48	0.979	1.01	1.03	1.102
MSPE	48	126.008	248.478	549.888	515.877
R2	48	0.378	0.372	0.626	0.606

Fig. 16: 48 Length Metric

48 saatlik zaman diliminde tahminler daha kısa vadeli olduğundan, modellerin başarı oranları daha yüksektir. Bu dönemdeki tahminlerde daha az belirsizlik ve daha net veriler olduğundan, modellerin doğruluğunun arttığı görülmektedir.

Metric	Length	Autoformer	Reformer	Informer	Vanilla
MSE	24	0.111	0.121	0.067	0.06
MAE	24	0.254	0.289	0.206	0.193
RMSE	24	0.334	0.348	0.259	0.245
MAPE	24	0.974	0.783	0.795	0.778
MSPE	24	385.553	211.751	328.618	281.595
R2	24	0.647	0.616	0.787	0.81

Fig. 17: 24 Length Metric

24 saatlik zaman diliminde yapılan tahminler, en kısa süreli tahminlerdir ve genellikle en doğru sonuçları verir.

III. SONUÇ VE DEĞERLENDİRME

Bu projede, petrol fiyat tahmini amacıyla beş farklı transformer modeli kullanılarak çeşitli deneyler gerçekleştirilmiştir. Modellerin seçiminde zaman serisi verilerinin özellikleri dikkate alınmış ve her bir modelin performansı belirli metriklerle ölçülmüştür. Projeye dahil edilen modeller, Autoformer, Informer, Reformer, Vanilla Transformer ve ETT olarak belirlenmiştir. Her bir modelin eğitimi, modelin genel özelliklerine uygun hiperparametreler kullanılarak gerçekleştirilmiştir.

İlk olarak, veri kümesi temizlenmiş ve gerekli sütunlar (tarih, açılış, kapanış, yüksek, düşük fiyatlar, teknik göstergeler vb.) seçilmiştir. Veri, eğitim ve test olmak üzere iki ayrı küme olarak bölünmüştür. Eğitim verisi modelin öğrenmesini sağlarken, test verisi tahminlerin doğruluğunu değerlendirmeye yarayacaktır.

Projede beş farklı transformer modeli uygulanmıştır. Her bir modelin eğitim süreci, eğitim verisi üzerinde optimizasyon yapılacak şekilde yapılandırılmıştır. Hiperparametreler, modelin eğitim başarısını artıracak şekilde belirlenmiştir.

Modellerin başarısı, Ortalama Mutlak Hata (MAE), Ortalama Yüzde Hata (MAPE), Kök Ortalama Kare Hata (RMSE), Ortalama Kare Hata (MSE) ve R-Kare (R^2) gibi performans metrikleri kullanılarak ölçülmüştür.

Deneylerin sonuçları, her modelin doğruluğu ve genel performansı açısından farklılıklar göstermiştir. Autoformer ve Informer modelleri, kısa vadeli tahminlerde daha başarılı olmuş ve genellikle hızlı sonuçlar üretmiştir. Bu modellerin özellikle veri sıklığının yüksek olduğu dönemlerde daha stabil sonuçlar verdiği görülmüştür. ETT modeli, uzun vadeli bağımlılıkları doğru tahmin etmiştir. Bu model, diğer transformer modellerine göre daha yüksek doğruluk sağlamış ve tahmin başarısı açısından en verimli sonuçları sunmuştur. Eğitim süresi daha uzun olsa da, elde edilen performans bunu telafi etmiştir.

Reformer ve Vanilla Transformer modelleri, genellikle daha düşük doğruluklarla sonuçlanmış, ancak eğitim süreleri diğer modellere göre daha kısa olmuştur. Bu, bu modellerin zaman serisi verilerinin karmaşıklığını ve uzun vadeli bağımlılıkları yeterince iyi öğrenemediğini, ancak daha hızlı öğrenme süreçleri sunduklarını göstermektedir. Bu durum, bu modellerin daha basit yapıları nedeniyle yüksek doğruluklu tahminler yapmada zorluk çekebileceğini, ancak hız açısından avantaj sağladığını ortaya koymaktadır.

IV. GELECEK ÇALIŞMALAR

Bu çalışmada kullanılan transformer tabanlı modellerin başarısı, petrol fiyat tahmininde güçlü sonuçlar elde edilmesini sağlamıştır. Ancak, her modelin belirli sınırlamaları ve iyileştirme alanları bulunmaktadır. Gelecek çalışmalarla aşağıdaki iyileştirmeler yapılabilir:

- 1) **Hiperparametre Optimizasyonu:** Model performansını artırmak için hiperparametre optimizasyonu üzerinde daha fazla çalışma yapılabilir. Özellikle daha fazla deneysel parametre ile hiperparametre ayarlamaları yapılması, doğruluk oranlarını yükseltebilir.
- 2) **Model Kombinasyonları:** Birden fazla modelin birleşimi veya ensembling yöntemleri

kullanılabilir. Bu yaklaşım, farklı modellerin güçlü yönlerini birleştirerek daha yüksek doğruluk elde edilmesine olanak tanıyabilir.

- 3) **Ek Veriler ve Göstergeler:** Kullanılan parametrelerin yanı sıra diğer ekonomik göstergeler, jeopolitik faktörler ve haber verileri gibi ek kaynaklardan elde edilen verilerin dahil edilmesi, modelin genel başarısını artırabilir.
- 4) **Farklı Zaman Dilimleri ve Frekanslar:** Mevcut model sadece günlük verilere dayanarak tahmin yapmaktadır. Gelecek çalışmalarda, daha kısa (saatlik) veya daha uzun (haftalık) zaman dilimlerinde tahmin yapacak modeller incelenebilir.

Bu iyileştirmeler, gelecekteki araştırmalarda daha doğru ve sağlam petrol fiyat tahmin modelleri geliştirilmesine katkı sağlayabilir.

KAYNAKLAR

- [1] G. Puniani, "Reformer: The Efficient and Overlooked Transformer," *Medium*, [Online]. Available: <https://medium.com/@gobindpuniani/reformer-the-efficient-and-overlooked-transformer-a3e9cd9136da>
- [2] L. A. Santos, "Vanilla Transformer," *GitHub*, [Online]. Available: https://github.com/leonardoaraujosantos/Transformers/blob/master/Vanilla_Transformer.ipynb
- [3] "A Vanilla Transformer Architecture," *ResearchGate*, [Online]. Available: https://www.researchgate.net/figure/A-vanilla-transformer-architecture_fig1_353517366
- [4] H. Zhou, "Autoformer," *GitHub*, [Online]. Available: <https://github.com/thuml/Autoformer?tab=readme-ov-file>
- [5] Hugging Face, "Time Series Transformer," *Hugging Face Documentation*, [Online]. Available: https://huggingface.co/docs/transformers/model_doc/time_series_transformer
- [6] H. Zhou, "Informer: Beyond Efficient Transformer for Time Series Forecasting," *GitHub*, [Online]. Available: <https://github.com/zhouhaoyi/Informer2020>
- [7] Selenium WebDriver, "Selenium WebDriver Projects", Available: [Projects | Selenium](https://selenium.dev/projects)
- [8] Yahoo Finance, Available: <https://finance.yahoo.com/>
- [9] Ahmet Demirel, "Selenium ile nasıl web scraping (veri kazıma) yapılır ?", Available: [Selenium ile nasıl web scraping \(veri kazıma\) yapılır ? | by Ahmet Demirel | AhmetDemirel-blog | Medium](https://ahmetdemirel.com/selenium-ile-nasil-web-scraping-veri-kazima-yapilir/)
- [10] GeeksforGeeks, "Importerror: Cannot Import Name In Python", Available: [Importerror: Cannot Import Name In Python - GeeksforGeeks](https://www.geeksforgeeks.org/import-error-cannot-import-name-in-python/)
- [11] Muthukadan, B. (2011-2024). *Installation: Selenium Python Bindings 2 documentation*. Read the Docs. Retrieved November 14, 2024, from <https://selenium-python.readthedocs.io/installation.html>

[12]