1. **Giriş**

Bu proje, "Yapay Zekâ Destekli Veri Şifreleme ve Sıkıştırma" konusuna odaklanmaktadır. Temel amaç, yapay zekâ (YZ) tekniklerinin, veri güvenliğini artırma ve veri boyutunu azaltma süreçlerindeki potansiyelini araştırmak, uygulamak ve değerlendirmektir. Proje, YZ'nin geleneksel yöntemlere kıyasla nasıl daha adaptif, verimli ve güvenli çözümler sunabileceğini ortaya koymayı hedeflemektedir. Veri hacminin katlanarak arttığı günümüz dünyasında hem güvenliğin hem de depolama/iletim verimliliğinin eş zamanlı olarak iyileştirilmesi kritik bir ihtiyaçtır ve bu proje bu ihtiyaca YZ perspektifinden bir çözüm aramaktadır.

2. **Araştırma**

Proje kapsamında öncelikle kapsamlı bir literatür araştırması yapılacaktır. Bu araştırma; modern kriptografi (örneğin, homomorfik şifreleme, kuantum sonrası kriptografi temelleri), bilgi teorisi, kayıplı/kayıpsız veri sıkıştırma algoritmaları (Huffman, Lempel-Ziv, JPEG, MP3 vb.) ve ilgili YZ yaklaşımları (derin öğrenme, denetimli/denetimsiz öğrenme, pekiştirmeli öğrenme, üretken çekişmeli ağlar) üzerine odaklanacaktır. Özellikle, YZ'nin şifreleme anahtarı üretimi/yönetimi, kriptanaliz, veri içeriğine duyarlı adaptif sıkıştırma ve hatta şifreli veri üzerinde işlem yapma gibi alanlardaki mevcut uygulamaları ve teorik çalışmaları incelenecektir.

3. **Yöntem ve Karşılaştırma**

Bu projede, geleneksel şifreleme (AES, RSA vb.) ve sıkıştırma (Huffman, LZW vb.) yöntemleri ile YZ tabanlı potansiyel yeni yaklaşımlar veya hibrit modeller karşılaştırılacaktır. Geliştirilecek veya araştırılacak YZ yöntemleri arasında, veri örüntülerini öğrenerek daha iyi sıkıştırma oranları sunan sinir ağları veya güvenlik parametrelerini dinamik olarak ayarlayan pekiştirmeli öğrenme ajanları bulunabilir. Yöntemlerin karşılaştırılmasında kullanılacak temel performans metrikleri; şifreleme için güvenlik seviyesi (örneğin, anahtar uzayı büyüklüğü, bilinen ataklara karşı direnç), sıkıştırma için elde edilen oran, her iki işlem için de hesaplama süresi (gecikme), kaynak kullanımı (CPU, bellek) ve farklı türdeki verilere (metin, görüntü, yapısal veri vb.) uyum sağlama yeteneği olacaktır.

4. **Veri Seti ve Uygulama**

YZ modellerinin eğitimi ve geliştirilen yöntemlerin test edilmesi için çeşitli veri setleri kullanılacaktır. Bu veri setleri, projenin odaklanacağı uygulama alanlarına göre seçilecek olup metin belgeleri, görüntüler, ses dosyaları veya yapısal veritabanı örneklerini içerebilir. Potansiyel uygulama senaryoları arasında; kaynakları kısıtlı IoT cihazları için hafif ve güvenli iletişim protokolleri geliştirmek, bulut depolama sistemlerinde maliyeti ve güvenliği optimize etmek, hassas verilerin (sağlık kayıtları, finansal bilgiler) güvenli arşivlenmesi veya büyük veri analitiği platformlarında veri ön işleme adımlarını iyileştirmek yer almaktadır. Proje kapsamında, seçilen bir veya birkaç senaryo için Python programlama dili ve PyTorch gibi YZ kütüphaneleri kullanılarak Proof-of-Concept prototipleri geliştirilebilir. Performans metriklerine göre en başarılı kabul edilen yöntem C++ dilinde tekrar yazılır ve kullanıma alınma öncesinde stres testleri yapılır.

5. **Beklenen Sonuç**

Projenin sonunda, YZ'nin veri şifreleme ve sıkıştırma alanlarındaki etkinliğini ve potansiyelini detaylandıran kapsamlı bir teknik rapor sunulması beklenmektedir. Bu rapor, yapılan literatür araştırmasının özetini, önerilen veya incelenen YZ yöntemlerinin açıklamalarını, geleneksel yöntemlerle yapılan karşılaştırmalı performans analizlerini ve geliştirilen prototiplerin sonuçlarını içerecektir. Raporun, YZ tabanlı yaklaşımların hangi koşullar altında geleneksel yöntemlere üstünlük

sağlayabileceği, bu yaklaşımların pratik uygulama zorlukları (örneğin, model karmaşıklığı, eğitim süresi, açıklanabilirlik eksikliği) ve gelecek araştırma yönleri hakkında bilgiler sunması hedeflenmektedir.

Rapora ek olarak, proje süresince yazılan tüm kaynak kodların eksiksiz bir şekilde teslim edilmesi beklenmektedir. Teslim edilecek kodların, yapılan çalışmaları, kullanılan algoritmaları ve mantıksal akışı net bir şekilde anlamayı sağlayacak düzeyde, yeterli yorum satırları ile detaylıca açıklanmış olması gerekmektedir. Ayrıca, kodların başka bir geliştirici tarafından veya farklı bir zamanda minimum eforla sorunsuz bir şekilde tekrar çalıştırılabilir (reproducible) olması ve bağımlılıklarının (kütüphaneler, veri setleri vb.) açıkça belirtilmiş olması beklenmektedir.