1. **GİRİŞ VE AMAÇ**

**YAPAY ZEKA VE VERİ MÜHENDİSLİĞİ**

I**. Giriş**

*Yapay Zekâ ve Veri Mühendisliği: Tanım ve Önemi;*

Yapay zekâ ve veri mühendisliği günümüz teknolojisinin en temel taşlarından birini oluşturuyor ve pek çok açıdan derinlemesine incelenmeye değer bir konudur.

Yapay zekâ, bilgisayar sistemlerine insan zekâsına benzer yetenekler kazandırmayı hedefleyen bir alan olarak tanımlanabilir. Bu, genellikle öğrenme, karar verme, problem çözme ve dil anlama gibi süreçleri içerir. Yapay zekâ, makine öğrenimi ve derin öğrenme gibi alt alanlara sahiptir. Makine öğrenimi, bir algoritmanın veri setlerini analiz ederek modeller oluşturmasına ve gelecekteki verileri tahmin etmesine olanak tanırken, derin öğrenme, yapay sinir ağları aracılığıyla karmaşık desenleri ve ilişkileri tanımlamak için kullanılır.

Diğer yandan, veri mühendisliği, büyük miktarda verinin toplanması, depolanması, işlenmesi ve analiz edilmesi gibi süreçleri içerir. Bu, veri tabanları, veri depolama sistemleri, veri akışı yönetimi ve veri analitiği gibi alt alanları kapsar. Veri mühendisliği, büyük veri setlerinden anlamlı bilgiler elde etmek ve bu bilgileri karar alma süreçlerine dönüştürmek için önemlidir.

Yapay zekâ ve veri mühendisliği bir araya geldiğinde, büyük veri setlerinin analizi, öngörülebilir modellerin oluşturulması, otomatik karar verme süreçlerinin iyileştirilmesi ve işletmelerin verilerini daha etkin bir şekilde kullanması gibi pek çok fırsat ortaya çıkar. Örneğin, bir e-ticaret şirketi, müşteri davranışlarını analiz ederek kişiselleştirilmiş ürün önerileri sunabilir veya bir finans kuruluşu, dolandırıcılık tespiti için makine öğrenimi algoritmalarını kullanabilir.

Yapay zekâ ve veri mühendisliğinin önemi, işletmelerin rekabet avantajı elde etmelerine, operasyonel verimliliği artırmalarına ve müşteri deneyimini iyileştirmelerine olanak tanır. Ayrıca, sağlık, ulaşım, güvenlik, enerji ve daha birçok endüstride büyük etkileri vardır.

Ancak, bu teknolojilerin kullanımıyla ilgili bazı zorluklar da vardır. Bunlar, veri gizliliği, etik sorunlar, algoritmik adalet ve iş gücü dönüşümü gibi konuları içerebilir. Bu nedenle, yapay zekâ ve veri mühendisliği alanında çalışan profesyonellerin sadece teknik bilgiye değil, aynı zamanda etik ve sosyal sorumluluk konularına da dikkat etmeleri önemlidir.

Sonuç olarak, yapay zekâ ve veri mühendisliği, günümüzde ve gelecekte teknolojik ilerlemenin anahtarını oluşturuyor ve bu alanlarda yapılan çalışmalar, birçok endüstride büyük değişimlere yol açıyor. Bu nedenle, bu alanlara olan ilgi ve yatırımın artması beklenmektedir.

1. **GENEL BİLGİLER**
   1. **Yapay Zeka Temelleri?**

Modern bilgisayarların ve yapay zekanın kurucularından biri olan Alan Turing (1950), “Turing testi” ile bir bilgisayarın akıllı davranışının, bilişle ilgili görevlerde insan düzeyinde performans elde etme yeteneği olduğu gerçeğini ortaya koymuştur (Batal, 2016). Yapay zekaya olan ilgi ise 1980'li ve 1990'lı yıllarda artış göstermiştir. Özellikle sağlık hizmetlerinde farklı klinik ortamlarda yazılım literatüründeki; bulanık uzman sistemler, Bayes ağları, yapay sinir ağları ve hibrit akıllı sistemler gibi yapay akıllı teknikler kullanılmaktadır. Yapay zeka araştırmalarına 2016 yılında yapılan en büyük yatırımın diğer sektörlere kıyasla sağlık uygulamaları alanında olduğu saptanmıştır (Prim, 2016).

Yapay zekanın tarihçesi dönemlere ayrılarak değerlendirildiğinde genel görüş dört farklı dönemde incelenmektedir. Bu dönemler sınıflandırıldığında:

Tarih Öncesi Dönem: Binlerce yıl önce Yunan mitolojisinde Rüzgar Tanrısı zannedilen Daedelus’un "yapay-insan" denemesi olduğu bilinmektedir.

Karanlık Dönem (1965-1970): Bilgisayar uzmanları, düşünen bir mekanizma geliştirmeye çalışmış ve sadece verileri yükleyerek akıllı bilgisayarlar yapmayı hedeflemişlerdir. Bu dönemde çok az gelişmeler elde edilmiş ve kısa süreli bir bekleme dönemine girilmiştir.

Rönesans Dönemi (1970-1975): Bilim insanları, hastalıkların teşhisi gibi çok önemli sistemler geliştirmişlerdir. Rönesans dönemi, sonraki dönemlere ışık tutacak ve yapay zeka çalışmalarına ivme kazandıracak çalışmaların yapıldığı dönemdir.

Ortaklık Dönemi (1975-1980): Yapay zeka ile ilgilenen bilim insanları, dil ve psikoloji gibi diğer bilim dallarından da yararlanmaya başlanıldığı dönemdir. Girişimcilik Dönemi (1980-?): Yapay zeka, geliştirildikleri laboratuvarların dışına çıkarılarak, gündelik hayatın içerisinde kullanılmaya ve geliştirilmeye başlanmıştır. Bu dönem, günümüzde halen devam etmektedir (Karaduman, 2019).

Yapay zeka veya bilimsel adıyla mantıksal analiz robotunun bir tanımında bilgisayarın veya bilgisayar kontrolündeki robotun insana özgü niteliklerden olan akıl yürütme, anlam çıkartma ve geçmiş deneyimlerden öğrenme gibi yüksek zeka gerektiren faaliyetleri, insan zekasının çalışma modelini örnek alarak gerçekleştirmesi olarak ifade edilebilir.

Birçok bilim insanı yapay zeka ile ilgili çok çeşitli tanımlarda bulunmuştur. Tanımlardan bazıları şu şekildedir (Kapucu, 2014);

Genesereth ve Nilsson’a göre yapay zeka; "Akıllı davranış üzerine bir çalışmadır. Ana hedefi, doğadaki varlıkların akıllı davranışlarını yapay olarak üretmeyi amaçlayan bir kuram oluşturmasıdır.” şeklinde ifade edilirken, Tesler’e göre bu tanım; "Şu ana kadar yapılmayanlardır." Axe göre yapay zeka; "Akıllı programları hedefleyen bir bilimdir." Slage’ye göre ise; "İnsanların yaptıklarını bilgisayarlara yaptırabilme çalışmasıdır." İfadelerini kullandıkları belirtilmiştir.

İnsan aklına benzetilmiş yapay beyin ve yapay bilinç sistemi, insan aklı kadar esnek ve duyguları olan bir karar alma mekanizması, uzman sistemler kadar yetkin bir bilgi birikimi ile yapay zeka, gelecekte insan zekasına bir alternatif oluşturabileceği düşünülmektedir (Batal, 2016).

***2.1.1 yapay zeka nedir?***

Yapay zeka (YZ) günümüzde bilgisayar biliminin ve teknolojinin en heyecan verici alanlarından biridir. Temel olarak, insan benzeri zeka ve davranışları modellemeyi amaçlar. Bu, bilgisayar sistemlerinin öğrenme, problem çözme, karar verme, dil anlama, algılama ve hatta duygusal tepkiler gibi insan yeteneklerini simüle etmeye çalışması anlamına gelir.

Yapay zeka alanı, iki ana yaklaşıma dayanır: sembolik yapay zeka ve bağlantılı yapay zeka. Sembolik yapay zeka, semboller ve mantıksal kurallar kullanarak zeka göstermeyi amaçlar. Bu yaklaşım, uzman sistemler, dil işleme ve bilgi temelli sistemler gibi alanlarda kullanılır. Bağlantılı yapay zeka ise, sinir ağları ve derin öğrenme gibi yapay sinir ağı modellerini kullanarak öğrenme ve davranış sergilemeyi hedefler.

Yapay zeka birçok endüstride çeşitli uygulamalar bulmuştur:

1. \*\*Otomasyon ve Üretim\*\*: Fabrika otomasyonu, robotik sistemler ve lojistik yönetimi gibi alanlarda yapay zeka, iş süreçlerini optimize etmek ve verimliliği artırmak için kullanılır.

2. \*\*Sağlık Hizmetleri\*\*: Tıbbi teşhis ve tedavi planlaması, görüntü işleme, genetik analiz gibi alanlarda yapay zeka, hastalıkların erken teşhisi ve tedavi süreçlerinin iyileştirilmesi için kullanılır.

3. \*\*Ulaşım ve Taşımacılık\*\*: Sürücüsüz araçlar, trafik optimizasyonu, otonom dronlar gibi alanlarda yapay zeka, güvenli ve etkili taşımacılık sistemleri geliştirmek için kullanılır.

4. \*\*Finansal Hizmetler\*\*: Dolandırıcılık tespiti, portföy yönetimi, otomatik ticaret sistemleri gibi alanlarda yapay zeka, finansal karar alma süreçlerini optimize etmek için kullanılır.

5. \*\*Eğitim ve Öğrenme\*\*: Bireyselleştirilmiş öğrenme materyalleri, öğrenci performans analizi, sanal sınıflar gibi alanlarda yapay zeka, eğitim süreçlerini iyileştirmek için kullanılır.

Yapay zeka teknolojileri, veri bilimi, makine öğrenimi, derin öğrenme, doğal dil işleme ve görüntü işleme gibi alt alanlarda sürekli olarak gelişmektedir. Bu gelişmeler, yapay zekanın daha karmaşık ve insan benzeri görevleri yerine getirebilme yeteneğini artırmaktadır. Ancak, etik ve güvenlik konuları da bu teknolojinin yaygın kullanımını düzenleyen önemli faktörlerdir.

***2.1.2 yapay zeka tarihçesi***

Yapay zeka (YZ) tarihçesi oldukça zengin ve karmaşıktır. İşte yapay zekanın önemli kilometre taşlarından bazıları:

1. \*\*Alan Turing ve Turing Testi (1950'ler):\*\* Alan Turing, "Bir Bilgisayarın Düşünebilir mi?" adlı makalesinde, bir bilgisayarın insan gibi davranıp davranamayacağını ölçmek için Turing Testi'ni önerdi. Bu test, bir bilgisayarın insanla konuşarak onu gerçek bir insan gibi kandırıp kandıramayacağını değerlendirir.

2. \*\*McCulloch-Pitts Hücresi (1943):\*\* Warren McCulloch ve Walter Pitts, biyolojik sinir hücrelerinin basit bir matematiksel modelini tanımladılar. Bu model, yapay sinir ağlarının temelini oluşturdu.

3. \*\*Dartmouth Konferansı (1956):\*\* John McCarthy, Marvin Minsky, Nathaniel Rochester ve Claude Shannon, yapay zeka terimini ortaya çıkardıkları ve yapay zeka araştırmalarını koordine etmek için Dartmouth College'da bir konferans düzenlediler.

4. \*\*Başlangıçtaki İlgili Alanlar (1950'ler ve 1960'lar):\*\* İlk yapay zeka çalışmaları, sembolik yapay zeka ve uzman sistemler gibi alanlarda yoğunlaştı. Dönemin bilim insanları, dil işleme, bilgi temsili ve problem çözme gibi konular üzerine çalıştılar.

5. \*\*Başarısızlık Dönemi (1970'ler ve 1980'ler):\*\* İlk yapay zeka atılımlarının ardından, beklenen ilerleme gerçekleşmedi ve yapay zeka araştırmalarında bir durgunluk dönemi yaşandı. Bu dönemde, yapay zeka araştırmaları için yeterli veri ve hesaplama gücü bulunmuyordu.

6. \*\*Yeniden Canlanma ve Makine Öğrenmesi (1990'lar ve 2000'ler):\*\* Yapay zeka araştırmaları, özellikle makine öğrenmesi ve derin öğrenme gibi alanlarda yeniden canlandı. Artan hesaplama gücü ve büyük veri setlerinin kullanılabilir hale gelmesi, yapay zeka alanında önemli ilerlemelere yol açtı.

7. \*\*Modern Yapay Zeka (2010'lar ve Sonrası):\*\* Günümüzde, yapay zeka birçok alanda uygulanmaktadır. Otomasyon, sağlık, ulaşım, finans ve eğitim gibi birçok sektörde yapay zeka teknolojileri kullanılmaktadır. Derin öğrenme ve doğal dil işleme gibi alanlardaki ilerlemeler, yapay zekanın daha karmaşık ve insan benzeri görevleri yerine getirebilmesini sağlamıştır.

***2.1.3 temel kavramlar ve yaklaşımlar***

Yapay zeka (YZ) alanında temel kavramlar ve yaklaşımlar oldukça çeşitlidir. İşte yapay zekanın ana kavramlarından bazıları:

1. \*\*Makine Öğrenmesi (ML):\*\* Makine öğrenmesi, bilgisayar sistemlerinin veriden öğrenme yeteneği kazanmasını sağlayan bir yapay zeka alt dalıdır. Makine öğrenmesi, deneyimden öğrenme, desen tanıma, sınıflandırma ve tahminleme gibi teknikleri içerir.

2. \*\*Derin Öğrenme (DL):\*\* Derin öğrenme, yapay sinir ağları adı verilen derin mimarilere dayanan bir makine öğrenmesi alt dalıdır. Derin öğrenme, karmaşık veri yapılarını analiz etmek ve anlamak için kullanılır ve son yıllarda birçok alanda büyük başarı elde etmiştir.

3. \*\*Doğal Dil İşleme (NLP):\*\* Doğal dil işleme, insan dilini anlamak, yorumlamak ve üretmek için bilgisayar sistemlerini kullanma sürecidir. NLP, metin sınıflandırma, duygu analizi, metin oluşturma ve çeviri gibi çeşitli uygulamalara sahiptir.

4. \*\*Bilgisayar Görüşü (CV):\*\* Bilgisayar görüşü, bilgisayarların dijital görüntülerle çalışmasını sağlayan bir yapay zeka alt dalıdır. CV, nesne tanıma, yüz tanıma, video izleme ve medikal görüntüleme gibi birçok uygulama alanına sahiptir.

5. \*\*Yapay Sinir Ağları (ANN):\*\* Yapay sinir ağları, biyolojik sinir ağlarının bir modelini oluşturan bir yapay zeka teknolojisidir. ANN'ler, derin öğrenme ve makine öğrenmesinin temelini oluşturur.

6. \*\*Uzaktan Denetim (Reinforcement Learning):\*\* Uzaktan denetim, bir ajanın çevresini keşfederek ve etkileşimlerden geri bildirim alarak öğrenme sürecini ifade eder. Bu yaklaşım, oyunlar, robotik ve otomasyon gibi alanlarda kullanılır.

7. \*\*Semantik Analiz (Semantic Analysis):\*\* Semantik analiz, metinlerin anlamını anlama sürecidir. Bu, metinleri anlamsal olarak işlemek ve içerdikleri bilgiyi anlamak için kullanılır.

Bu kavramlar ve yaklaşımlar, yapay zeka alanında çalışan araştırmacılar ve mühendisler tarafından geniş çapta kullanılmaktadır. Her biri, farklı uygulamalarda kullanılarak insan benzeri zeka ve davranışların modellenmesine katkıda bulunur.

**3. MAKİNE ÖĞRENİMİ**

**3.1 Makine Öğrenimi?**

***3.1.1 makine öğrenimi nedir?***

Makine öğrenimi (MO), bilgisayar sistemlerinin deneyimlerinden öğrenme yeteneği kazanmasını sağlayan bir yapay zeka alt dalıdır. Temel olarak, bir makine öğrenimi algoritması, belirli bir görevi yerine getirmek için verilerden öğrenir ve bu göreve uygun bir model oluşturur.Makine öğrenimi algoritmaları, genellikle iki ana kategoride incelenir: denetimli öğrenme ve denetimsiz öğrenme.

1. \*\*Denetimli Öğrenme (Supervised Learning):\*\* Denetimli öğrenme, girdi ve çıktı arasındaki ilişkiyi öğrenmek için etiketlenmiş verilerle eğitilmiş algoritmaları kullanır. Bu tür algoritmalar, belirli bir örnek girdisi verildiğinde doğru çıktıyı üretmek için modeli eğitmek için kullanılır. Denetimli öğrenme, sınıflandırma ve regresyon gibi görevler için kullanılır. Örnek algoritmalar arasında destek vektör makineleri (SVM), karar ağaçları, doğrusal regresyon ve yapay sinir ağları bulunur.

2. \*\*Denetimsiz Öğrenme (Unsupervised Learning):\*\* Denetimsiz öğrenme, etiketlenmemiş verilerle çalışır ve veri içindeki kalıpları ve yapılardan otomatik olarak öğrenmeye çalışır. Bu tür algoritmalar, veri kümesindeki doğal gruplamaları (kümeleme), veri boyutunu azaltmayı (boyut azaltma) veya veri dağılımını incelemeyi (görüntüleme) içerebilir. Örnek algoritmalar arasında K-ortalama kümeleme, PCA (Principal Component Analysis) ve doğrusal olmayan boyut azaltma yöntemleri yer alır.

Makine öğrenimi, birçok uygulama alanında kullanılır, örneğin:

- Otomatik sürüş ve otonom araçlar

- Sağlık hizmetleri ve tıbbi teşhis

- Finansal tahminler ve risk yönetimi

- Doğal dil işleme ve metin analizi

- Görsel tanıma ve bilgisayarlı görüş

- Kişisel öneri sistemleri ve reklamcılık

Makine öğrenimi, büyük veri setlerinden anlamlı bilgiler elde etmek, karar alma süreçlerini optimize etmek ve karmaşık problemleri çözmek için güçlü bir araçtır.

***3.1.2 makine öğrenimi algoritmaları***

Makine öğrenimi alanında birçok farklı algoritma bulunmaktadır, her biri belirli bir problemi çözmek veya belirli bir görevi yerine getirmek için kullanılır. İşte bazı temel makine öğrenimi algoritmaları:

1. \*\*Destek Vektör Makineleri (Support Vector Machines - SVM):\*\* Sınıflandırma ve regresyon problemleri için kullanılan bir algoritmadır. SVM, veri noktalarını sınıflandırmak için bir hiperdüzlem oluşturur ve bu hiperdüzlemi en iyi şekilde ayıran sınırlayıcıları belirler.

2. \*\*Karar Ağaçları (Decision Trees):\*\* Sınıflandırma ve regresyon problemleri için kullanılan bir ağaç yapısıdır. Karar ağaçları, veri kümesindeki özelliklere göre sıralanmış karar düğümlerinden oluşur ve sonuçları tahmin etmek için kullanılır.

3. \*\*Doğrusal Regresyon (Linear Regression):\*\* Bir bağımsız değişkenin bir veya daha fazla bağımlı değişken üzerindeki etkisini incelemek için kullanılan bir regresyon analizi yöntemidir. Doğrusal regresyon, bir doğrusal model oluşturarak bağımlı değişkenin tahmin edilmesine yardımcı olur.

4. \*\*K-Ortalama Kümeleme (K-Means Clustering):\*\* Veri noktalarını belirli sayıda küme veya kümeye gruplamak için kullanılan bir kümeleme algoritmasıdır. K-ortalama, veri noktalarını rastgele kümelere atar ve ardından merkezleri güncelleyerek kümeleme işlemini tekrar eder.

5. \*\*Doğrusal Olmayan Boyut Azaltma (Non-linear Dimensionality Reduction):\*\* Veri boyutunu azaltmak ve verinin özünü daha az boyutta temsil etmek için kullanılan bir dizi teknik vardır. Örneğin, PCA (Principal Component Analysis) ve t-SNE (t-distributed Stochastic Neighbor Embedding) gibi algoritmalar, veri boyutunu azaltmak için kullanılabilir.

6. \*\*Rastgele Ormanlar (Random Forests):\*\* Karar ağaçlarının bir ensemble versiyonudur. Rastgele ormanlar, birçok karar ağacını eğiterek verileri sınıflandırmak veya regresyon yapmak için kullanılır. Her bir ağaç farklı bir alt kümesiyle eğitilir ve sonuçlar ortalamalanır.

7. \*\*Derin Sinir Ağları (Deep Neural Networks - DNN):\*\* Yapay sinir ağlarının derin bir mimarisine sahip olan bu algoritma, büyük veri setlerinden karmaşık desenleri tanımak ve öğrenmek için kullanılır. Derin sinir ağları, görüntü tanıma, doğal dil işleme ve çeviri gibi alanlarda geniş bir kullanım alanına sahiptir.

Bu, sadece makine öğrenimi alanında kullanılan bazı temel algoritmaların bir örneğidir. Her bir algoritmanın avantajları, dezavantajları ve farklı uygulama alanları bulunmaktadır, bu nedenle hangi algoritmanın belirli bir problem için en uygun olduğunu belirlemek genellikle deneyim ve problem bağlamına bağlıdır.

***3.1.3 uygulama örnekleri***

Makine öğrenimi, çeşitli alanlarda birçok farklı uygulama örneği sunar. İşte bu uygulama örneklerinden bazıları:

* \*\*Sağlık Hizmetleri:\*\*

\*\*Tıbbi Teşhis:\* Makine öğrenimi algoritmaları, hastalıkların tanısında ve tedavi planlarının oluşturulmasında kullanılabilir. Örneğin, derin öğrenme modelleri, radyoloji görüntülerinde kanser tespiti yapabilir.

* + \*\*Hasta Monitörizasyonu:\*\* Makine öğrenimi, hastaların sağlık verilerini analiz ederek kritik durumları önceden tespit edebilir ve sağlık hizmeti sağlayıcılarına uyarılar gönderebilir.
* \*\*Finansal Hizmetler:\*\*
  + \*\*Risk Değerlendirmesi:\*\* Bankalar ve sigorta şirketleri, müşteri profillerini ve geçmiş finansal işlemleri analiz ederek kredi riskini değerlendirebilir.
  + \*\*Hisse Senedi Tahminleri:\*\* Makine öğrenimi, hisse senedi piyasalarındaki trendleri ve fiyat hareketlerini tahmin etmek için kullanılabilir.
* \*\*Perakende ve E-Ticaret:\*\*
  + \*\*Kişiselleştirilmiş Öneriler:\*\* Büyük perakendeciler ve e-ticaret platformları, müşteri tercihlerini ve geçmiş alışveriş alışkanlıklarını analiz ederek kişiselleştirilmiş ürün önerileri sunabilir.
  + \*\*Stok Yönetimi:\*\* Makine öğrenimi, talep tahminlemesi ve envanter optimizasyonu için kullanılabilir, böylece perakendeciler stok seviyelerini optimize edebilirler.
* \*\*Otomotiv ve Ulaşım:\*\*
  + \*\*Sürücüsüz Araçlar:\*\* Makine öğrenimi, sürücüsüz araçların çevresel koşulları algılaması ve trafikte güvenli bir şekilde hareket etmesi için kullanılabilir.
  + \*\*Trafik Yönetimi:\*\* Şehirler, makine öğrenimi algoritmalarını kullanarak trafik akışını izleyebilir ve optimize edebilirler.
* \*\*Görüntü ve Video İşleme:\*\*
  + \*\*Nesne Tanıma:\*\* Makine öğrenimi algoritmaları, görüntülerdeki nesneleri tanımlayabilir ve sınıflandırabilir.
  + \*\*Video İzleme:\*\* Güvenlik kameralarından elde edilen video verileri, makine öğrenimi algoritmalarıyla analiz edilerek şüpheli aktiviteler tespit edilebilir.
* \*\*Doğal Dil İşleme:\*\*
  + \*\*Metin Sınıflandırma:\*\* Makine öğrenimi, metin verilerini analiz ederek duygu analizi yapabilir veya belirli kategorilere sınıflandırabilir.
  + \*\*Metin Üretimi:\*\* Derin öğrenme modelleri, makine öğrenimi ile metin oluşturabilir ve otomatik metin yazılımı geliştirebilir.

**4.DERİN ÖĞRENME**

**4.1 Derin Öğrenme?**

***4.1.1 derin öğrenme temelleri***

Derin öğrenme, yapay sinir ağları adı verilen derin bir mimariye dayanan bir makine öğrenimi alt dalıdır. Derin öğrenme, karmaşık desenleri tanımlamak ve öğrenmek için büyük veri setlerinden öğrenme yeteneği kazanır. İşte derin öğrenmenin temel bileşenleri ve işleyişi:

1. \*\*Yapay Sinir Ağları (Artificial Neural Networks - ANN):\*\* Derin öğrenmenin temeli, biyolojik sinir ağlarını taklit eden yapay sinir ağlarıdır. Yapay sinir ağları, birbirine bağlı katmanlardan oluşur ve her katman birbirine bağlı yapay sinir hücrelerinden oluşur. Bu sinir ağları, veri üzerinde karmaşık işlemler gerçekleştirebilir.

2. \*\*Katmanlar (Layers):\*\* Bir yapay sinir ağındaki her bir katman, bir dizi yapay sinir hücresinden oluşur. Tipik olarak, bir yapay sinir ağı üç tip katmandan oluşur: giriş katmanı, gizli katmanlar ve çıkış katmanı. Gizli katmanlar arasındaki sinir hücrelerinin sayısına ve düzenlemesine göre, ağın "derinliği" belirlenir.

3. \*\*Ağırlıklar ve Biaslar (Weights and Biases):\*\* Her sinir hücresi, girişlerinden gelen sinyalleri ağırlıklandırır ve bir bias ekler. Ağırlıklar ve biaslar, sinir ağındaki bağlantıların gücünü ve katmanlar arasındaki bilgi aktarımını belirler.

4. \*\*Aktivasyon Fonksiyonları (Activation Functions):\*\* Aktivasyon fonksiyonları, sinir ağının her bir sinir hücresinin çıktısını hesaplamak için kullanılır. Sigmoid, ReLU (Rectified Linear Activation), tanh (hyperbolic tangent) gibi yaygın aktivasyon fonksiyonları kullanılabilir.

5. \*\*İleri Besleme (Feedforward):\*\* İleri besleme, sinir ağında girişten çıkışa doğru ilerleyen bir süreçtir. Giriş verileri giriş katmanına verilir ve ardından gizli katmanlardan geçerek çıkış katmanına ulaşır.

6. \*\*Geribesleme (Backpropagation):\*\* Geribesleme, ağın ürettiği çıktıların gerçek etiketlerle karşılaştırılarak hatanın geriye doğru yayılmasıdır. Bu hata, ağın içindeki ağırlık ve bias değerlerini güncellemek için kullanılır. Bu işlem, ağın tahmin yeteneğini artırmak için iteratif olarak tekrarlanır.

Derin öğrenme, büyük veri setlerinden karmaşık desenleri otomatik olarak öğrenmek için güçlü bir araçtır. Görüntü tanıma, doğal dil işleme, oyun oynama ve otomatik sürüş gibi birçok alanda başarıyla kullanılmıştır. Derin öğrenme, yapay zeka alanında önemli bir ilerleme kaydetmiştir ve birçok uygulama için vazgeçilmez bir araç haline gelmiştir.

***4.1.2 yapay sinir ağları ve derin öğrenme modelleri***

Yapay sinir ağları (YSA) ve derin öğrenme modelleri, makine öğrenimi alanında büyük bir etki yaratmıştır. İşte bu modellerin temel özellikleri ve bazı popüler türleri:

* \*\*Yapay Sinir Ağları (YSA):\*\* Yapay sinir ağları, biyolojik sinir ağlarından esinlenerek tasarlanmış matematiksel modeldir. YSA'lar, giriş verilerini işleyen bir dizi yapay sinir hücresinden oluşur ve genellikle üç tip katmandan oluşur: giriş katmanı, gizli katmanlar ve çıkış katmanı.
  + \*\*Besleme İleri Ağlar (Feedforward Neural Networks):\*\* Girişten çıkışa doğru ilerleyen ve hiçbir geri dönüş döngüsü bulunmayan yapay sinir ağlarıdır. Sınıflandırma ve regresyon gibi görevler için kullanılırlar.
  + \*\*Rekürrent Ağlar (Recurrent Neural Networks - RNN):\*\* Geri dönüş bağlantılarına sahip yapay sinir ağlarıdır. Bu bağlantılar, önceki adımlardan gelen bilgilerin ağırlıklı olarak hesaba katılmasını sağlar. Zaman serisi analizi, metin oluşturma ve dil modellemesi gibi görevlerde kullanılırlar.
* \*\*Evrişimli Sinir Ağları (Convolutional Neural Networks - CNN):\*\* Görüntü tanıma ve işleme alanında büyük bir başarı elde eden yapay sinir ağlarıdır. CNN'ler, konvolüsyon ve pooling katmanlarından oluşur ve görüntü verilerini işlemek için optimize edilmiştir. Nesne tanıma, yüz tanıma ve görüntü sınıflandırması gibi görevlerde kullanılırlar.
* \*\*Derin Doğrusal Ağlar (Deep Belief Networks - DBN):\*\* Gelişmiş bir tür yapay sinir ağıdır ve genellikle unsupervised learning (denetimsiz öğrenme) görevlerinde kullanılır. Derin öğrenme modellerinden biri olarak kabul edilir ve birbirine bağlı katmanlardan oluşur.
* \*\*Uyarlamalı Moment Estimasyonu (Adaptive Moment Estimation - Adam):\*\* Adam, makine öğrenimi ve derin öğrenme modellerinde sıkça kullanılan bir optimizasyon algoritmasıdır. Gradyan inişine dayanan bir optimizasyon algoritmasıdır ve büyük veri setleriyle etkili bir şekilde çalışır.

Bu yapay sinir ağları ve derin öğrenme modelleri, birçok uygulama alanında başarıyla kullanılmış ve yapay zeka alanında önemli bir ilerleme sağlamıştır. Görüntü tanıma, doğal dil işleme, oyun oynama ve otomatik sürüş gibi alanlarda büyük başarılar elde edilmiştir.

***4.1.3 görüntü ve metin işleme uygulamaları***

Görüntü ve metin işleme, yapay zeka ve derin öğrenme alanlarında önemli uygulama alanlarına sahiptir. İşte bu alanlarda yaygın olarak kullanılan bazı uygulamalar:

* + - \*\*Görüntü İşleme Uygulamaları:\*\*
* \*\*Nesne Tanıma:\*\* Görüntülerdeki nesnelerin tanınması ve sınıflandırılması. Örneğin, trafik ışıklarının, araçların veya insanların tanınması.
* \*\*Yüz Tanıma:\*\* Görüntülerdeki yüzlerin tanınması ve tanımlanması. Bu, güvenlik sistemleri, biyometrik kimlik doğrulama ve fotoğraf etiketleme gibi alanlarda kullanılır.
* \*\*Görüntü Sınıflandırması:\*\* Görüntülerin belirli kategorilere sınıflandırılması. Örneğin, doğada, hayvanlarda veya trafik levhalarında görüntülerin tanınması.
* \*\*Görüntü Segmentasyonu:\*\* Görüntülerin piksel düzeyinde bölümlenmesi. Bu, bir görüntüdeki nesnelerin veya özelliklerin sınırlarını belirlemek için kullanılır.
* \*\*Görüntü Restorasyonu:\*\* Görüntülerdeki bozulmaların düzeltilmesi veya gürültülerin azaltılması. Örneğin, bulanıklık giderme veya parazit giderme.
  + - \*\*Metin İşleme Uygulamaları:\*\*
* \*\*Metin Sınıflandırma:\*\* Metin belgelerinin belirli kategorilere sınıflandırılması. Örneğin, spam filtreleme, duygu analizi veya konu etiketleme.
* \*\*Metin Özetleme:\*\* Uzun metin belgelerinin özetlenmesi veya kısaltılması. Örneğin, makalelerin veya raporların anahtar noktalarının belirlenmesi.
* \*\*Doğal Dil İşleme (NLP):\*\* Metin verilerinin işlenmesi ve anlaşılması. Bu, dil çevirisi, metin oluşturma, soru-cevap sistemleri ve konuşma tanıma gibi birçok uygulamayı içerir.
* \*\*Duygu Analizi:\*\* Metinlerdeki duygusal tonun belirlenmesi. Bu, sosyal medya analizi, ürün incelemelerinin analizi ve müşteri geri bildirimlerinin değerlendirilmesi gibi alanlarda kullanılır.
* \*\*Metin Madenciliği:\*\* Metin verilerindeki desenlerin ve ilişkilerin keşfedilmesi. Bu, kelime sıklıklarının analizi, kelime ilişkilerinin belirlenmesi ve metin kümelenmesi gibi teknikleri içerir.

Bu uygulamalar, görüntü ve metin verilerinin analizi, işlenmesi ve anlamlandırılması için kullanılan temel tekniklerdir. Her biri, yapay zeka ve derin öğrenme yöntemlerinin kullanıldığı çeşitli alanlarda büyük öneme sahiptir.

**5. VERİ MÜHENDİSLİĞİ**

**5.1 Veri Mühendisliği**

Veri mühendisliği, büyük veri sistemlerinin tasarımı, geliştirilmesi, uygulanması ve yönetilmesi süreçlerini içeren bir disiplindir. Temel olarak, veri mühendisleri, büyük miktarlardaki yapılandırılmış veya yapılandırılmamış verileri toplar, depolar, işler, analiz eder ve sunarlar. Bu süreçler, veri odaklı karar almayı ve işletme performansını geliştirmeyi desteklemek için kritiktir.

Veri mühendisliği, şu anahtar adımları içerir:

1. \*\*Veri Toplama:\*\* Farklı kaynaklardan (veri tabanları, uygulama günlükleri, sensörler, sosyal medya platformları, web servisleri vb.) gelen veriyi toplama sürecidir.

2. \*\*Veri Temizleme ve Dönüşüm:\*\* Toplanan verilerin kalitesini artırmak için veri temizleme ve dönüştürme işlemleri gerçekleştirilir. Bu adımda, eksik veya hatalı veriler düzeltilir, gereksiz veriler filtrelenir ve veri yapıları uygun formata dönüştürülür.

3. \*\*Veri Depolama:\*\* Verilerin güvenli, ölçeklenebilir ve erişilebilir bir şekilde saklanmasını sağlamak için uygun veri depolama sistemleri oluşturulur. Bu, ilişkisel veritabanları, NoSQL veritabanları, veri gölgeleme sistemleri ve bulut depolama hizmetleri gibi çeşitli teknolojileri içerebilir.

4. \*\*Büyük Veri İşleme:\*\* Büyük miktardaki verilerin etkili bir şekilde işlenmesi ve analiz edilmesi için dağıtık ve ölçeklenebilir hesaplama sistemleri tasarlanır ve uygulanır. Bu, Hadoop, Spark, Flink gibi açık kaynaklı çözümleri veya bulut tabanlı platformları içerebilir.

5. \*\*Veri Modelleme ve Analitik:\*\* Veri mühendisleri, veri ambarları oluşturur ve veri modelleme teknikleri kullanarak veri analizi ve raporlama işlemlerini gerçekleştirir. Bu, işletme ihtiyaçlarına uygun veri yapılarının tasarlanmasını, veri görselleştirme araçlarıyla analitik raporların hazırlanmasını ve veri tabanlı karar almayı destekler.

6. \*\*Veri Güvenliği ve Uyumluluk:\*\* Veri güvenliği standartlarının uygulanması ve veri gizliliğinin korunması, veri mühendislerinin önemli sorumluluklarındandır. Ayrıca, veri işleme ve depolama süreçlerinin yasal düzenlemelere uygunluğunun sağlanması da kritiktir.

Veri mühendisliği, günümüz iş dünyasında veri tabanlı karar alma süreçlerini güçlendirmek, işletmelerin rekabet avantajını artırmak ve veri odaklı inovasyonu teşvik etmek için önemli bir rol oynamaktadır. Bu nedenle, veri mühendisleri, hem teknik hem de işletme becerilerine sahip olmalıdır ve genellikle çok disiplinli ekiplerde çalışırlar.

***5.1.1 büyük veri ve veri kaynakları***

Büyük veri, hacim, çeşitlilik ve hız gibi özellikleriyle geleneksel veri işleme tekniklerinin sınırlarını zorlayan ve genellikle geleneksel veri depolama ve işleme sistemlerinin başa çıkamayacağı boyuttaki veri kümelerini tanımlar. Büyük veri, genellikle "3V" olarak adlandırılan üç temel özellik üzerinden tanımlanır:

1. \*\*Hacim (Volume):\*\* Büyük veri, trilyonlarca kayıt veya veri noktasını içerebilecek kadar büyük miktarda veriyi ifade eder. Bu veri, terabaytlar veya petabaytlar düzeyinde olabilir.

2. \*\*Çeşitlilik (Variety):\*\* Büyük veri, yapılandırılmış, yapılandırılmamış ve yarı yapılandırılmış veri türlerini kapsar. Yapılandırılmış veri, ilişkisel veritabanları gibi tablo biçiminde düzenlenmiş verileri içerirken, yapılandırılmamış veri, metin dosyaları, sosyal medya akışları, video ve ses gibi formatlarda olan verileri ifade eder.

3. \*\*Hız (Velocity):\*\* Büyük veri, hızlı bir şekilde oluşturulan ve işlenen veri akışlarını kapsar. Örneğin, sosyal medya platformlarından, sensör ağlarından veya internet trafiğinden sürekli olarak gelen veriler bu kategoriye girer.

Büyük veri genellikle çeşitli kaynaklardan gelir. İşte yaygın büyük veri kaynaklarından bazıları:

1. \*\*Sosyal Medya Verileri:\*\* Facebook, Twitter, Instagram gibi sosyal medya platformlarından üretilen veriler. Bu veriler, kullanıcı etkileşimleri, beğeniler, paylaşımlar ve yorumlar gibi çeşitli biçimlerde olabilir.

2. \*\*İnternet Trafiği Verileri:\*\* Web sitelerinin ve uygulamaların kullanıcı trafiği ve etkileşimleri hakkında bilgi sağlayan veriler. Bu veriler, web günlükleri, kullanıcı oturumları ve tıklama oranları gibi bilgileri içerebilir.

3. \*\*Sensör Verileri:\*\* Endüstriyel ekipmanlar, akıllı cihazlar, araçlar ve diğer nesneler tarafından üretilen sensör verileri. Bu veriler, sıcaklık, basınç, hız, konum gibi çeşitli parametreleri ölçer.

4. \*\*Satış ve Pazarlama Verileri:\*\* Şirketlerin satış işlemleri, müşteri ilişkileri yönetimi (CRM) verileri, pazarlama kampanyalarının etkileşimleri ve tüketici davranışları gibi bilgiler.

5. \*\*Biyo-informatik Verileri:\*\* Genomik veriler, proteomik veriler ve diğer biyolojik veri türleri. Bu veriler, sağlık, biyoloji ve tıp alanlarında genetik analizler ve tıbbi araştırmalar için kullanılır.

Bu büyük veri kaynakları, işletmelerin geniş veri kümelerinden değerli bilgiler çıkarmasına ve veriye dayalı kararlar almasına olanak tanır. Bununla birlikte, bu verilerin etkili bir şekilde yönetilmesi, depolanması, işlenmesi ve analiz edilmesi, veri mühendisliği disiplininin önemli bir parçasıdır.

***5.1.2 veri ön işleme ve temizleme örnekleri***

Veri ön işleme ve temizleme, veri analitiği ve makine öğrenimi projelerindeki önemli bir adımdır. Bu adımda, verilerin doğruluğunu, bütünlüğünü ve kullanılabilirliğini artırmak için çeşitli teknikler kullanılır. İşte yaygın olarak kullanılan bazı veri ön işleme ve temizleme teknikleri:

1. \*\*Eksik Veri İşleme:\*\* Veri setlerinde eksik veya boş değerler bulunabilir. Bu boş değerler, direkt olarak çıkarılabilir, eksik değerler tahmin edilerek doldurulabilir veya daha sofistike yöntemler kullanılarak tamamlanabilir.

2. \*\*Anormal Değerlerin Tespiti ve İşlenmesi:\*\* Anormal değerler (outlier'lar), genellikle yanlışlıkla veya hatalı ölçüm veya kayıt nedeniyle ortaya çıkar. Bu değerlerin tespit edilmesi ve işlenmesi önemlidir çünkü analiz sonuçlarını etkileyebilirler. Anormal değerleri tespit etmek için istatistiksel yöntemler veya eşik değerler kullanılabilir.

3. \*\*Veri Biçimlendirme ve Normalizasyon:\*\* Veri setlerindeki farklı özelliklerin farklı ölçeklerde olması yaygındır. Bu durum, bazı özelliklerin diğerlerine göre ağırlıklı olarak etkilenmesine neden olabilir. Veri biçimlendirme ve normalizasyon teknikleri kullanılarak, veri özellikleri arasındaki ölçek farklılıkları giderilir ve veri seti homojen hale getirilir.

4. \*\*Kategorik Değerleri Kodlama:\*\* Kategorik veriler (örneğin, cinsiyet, şehir, ürün kategorisi gibi) sayısal değerlerle ifade edilmedikleri için analiz için uygun değildir. Bu nedenle, kategorik verilerin sayısal değerlere dönüştürülmesi gerekir. Bu işlem, one-hot encoding, label encoding gibi tekniklerle gerçekleştirilir.

5. \*\*Tekrar Eden Verilerin Kaldırılması:\*\* Bazı veri setlerinde aynı bilgilerin farklı şekillerde veya tekrar eden şekillerde bulunması mümkündür. Bu gereksiz veri tekrarlarını kaldırmak, analiz sürecini optimize etmek ve bellek kullanımını azaltmak için önemlidir.

6. \*\*Özellik Seçimi ve Boyut Azaltma:\*\* Büyük boyutlu veri setlerinde, tüm özelliklerin analizinde pratik olmayabilir. Özellik seçimi ve boyut azaltma teknikleri kullanılarak, veri setinin boyutu azaltılabilir ve model performansı artırılabilir.

7. \*\*Veri Örneklemesi:\*\* Büyük veri setlerinde tamamını kullanmak maliyetli olabilir veya zaman alabilir. Bu nedenle, veri örnekleme teknikleri kullanılarak, daha küçük bir veri örneği üzerinde analiz yapılabilir.

Bu teknikler, veri ön işleme ve temizleme sürecinde yaygın olarak kullanılan ancak bunlarla sınırlı olmayan bazı örneklerdir. Veri mühendisleri ve veri bilimcileri, veri setlerinin kalitesini artırmak ve doğru sonuçlar elde etmek için bu teknikleri kullanırken, aynı zamanda spesifik projenin ihtiyaçlarına ve veri karakteristiklerine uygun teknikleri seçmek önemlidir.

Konuyla ilgili kısa örnek;

Tabloya bir sınıfın öğrenci notlarını içeren bir veri seti düşünelim:

| Öğrenci ID | Sınav 1 Notu | Sınav 2 Notu | Proje Notu | Final Notu |

|------------|--------------|--------------|------------|------------|

| 1 | 85 | 90 | 88 | 92 |

| 2 | 70 | 75 | 82 | 78 |

| 3 | 60 | 65 | 70 | 68 |

| 4 | 92 | 88 | 94 | 90 |

| ... | ... | ... | ... | ... |

Bu veri setini işlerken yapabileceğimiz bazı veri ön işleme adımları şunlar olabilir:

1. \*\*Eksik Veri İşleme:\*\* Veri setinde eksik veri olup olmadığını kontrol ederiz. Örneğin, herhangi bir öğrencinin notlarının eksik olduğu bir durum var mı diye bakarız. Eğer varsa, eksik verileri uygun bir şekilde doldurabiliriz veya eksik verilere sahip öğrencileri veri setinden çıkarabiliriz.

2. \*\*Anormal Değerlerin Tespiti ve İşlenmesi:\*\* Notlar arasında anormal derecede düşük veya yüksek değerler olup olmadığını kontrol ederiz. Örneğin, bir sınav notu 0 veya 100'den daha yüksekse, bu bir anormal değer olabilir. Bu tür anormal değerleri tespit edip uygun bir şekilde işleyebiliriz.

3. \*\*Veri Biçimlendirme ve Normalizasyon:\*\* Sınav notları gibi farklı özelliklerin farklı ölçeklerde olması yaygındır. Bu nedenle, notları 0 ile 100 arasında bir ölçekte tutabiliriz veya standartlaştırma veya normalizasyon tekniklerini kullanarak tüm özellikleri aynı ölçekte tutabiliriz.

4. \*\*Kategorik Değerleri Kodlama:\*\* Öğrenci ID gibi kategorik değerleri sayısal değerlere dönüştürmemiz gerekebilir. Bu, örneğin, her öğrenciye bir benzersiz bir sayı atayarak veya One-Hot Encoding gibi teknikler kullanarak yapılabilir.

5. \*\*Özellik Seçimi ve Boyut Azaltma:\*\* Eğer çok sayıda özellik varsa ve bazıları gereksiz ise, gereksiz özellikleri çıkarabiliriz veya boyut azaltma teknikleri kullanarak özellik sayısını azaltabiliriz.

6. \*\*Veri Örneklemesi:\*\* Veri seti çok büyükse, analiz süresini azaltmak için bir örneklem alabiliriz. Örneğin, rastgele seçilen belirli bir öğrenci alt kümesi üzerinde çalışabiliriz.

Bu adımları uygulayarak, veri setimizi analiz etmeye ve makine öğrenimi modelleri oluşturmaya hazır hale getirebiliriz. Bu sayede, doğru sonuçlar elde edebilir ve öğrencilerin akademik başarısını anlamak için veriye dayalı kararlar alabiliriz.

**6.VERİ MADENCİLİĞİ**

Sağlık sistemlerinin dijitalleşmesi ile klinikte büyük veri seline yol açtığı ve tıpta veri biliminin hızla büyümesine neden olduğu saptanmıştır (NelsonSanchez-Pinto et al., 2018).

Veriye ulaşım, ölçüm veya gözlem sonucunda bilgilerin kayıt altında tutulması ile gerçekleşebilmektedir. Veri madenciliğinin uygulanabilmesi için elde edilen büyük ölçekli verilerin işlenmesi, düzenlenmesi ve analiz edilmesi gerekmektedir (Erdem, 2008). Veri madenciliği kullanılarak farklı veriler arasında bağlantı ve etkileşimler keşfedilmektedir. Dolayısıyla veri kullanan kişilere bilgi vermek amacıyla veriler arasındaki doğru ilişkilerin kurulması sağlanmaktadır (Amin et al., 2019). Şekil 2.2.1’de veri madenciliği basamaklarını gösteren piramit verilmiştir.

Büyük veri yığınlarının bulunduğu dijital ortamlardaki veriler birbirleri ile ilişkilendirilmektedir. Bu verilerin gelecek ile ilgili tahminde bulunması ve buna bağlı olarak gerekli algoritmalara başvurması adımına veri madenciliği yöntemi adı verilir. Mevcut verilerin daha öncesinde keşfedilmemiş bilgilerini, belirli istatistiki yöntemler kullanarak ortaya çıkartılmasını veri madenciliği sağlamaktadır. Veri madenciliği, veriler arasındaki örüntüleri farklı algoritmaları kullanarak tespit etmeye çalışır (Yalçın, 2019). Günlük operasyonel ve stratejik karar vermeyi desteklemek için bilgi sistemleri; veri toplama, işleme ve dağıtma amaçların da kullanılmaktadır (Kayaalp ve Başarslan, 2018).

Büyük ham sağlık verileri koleksiyonunu, veri madenciliği yöntemi ile bilinçli karar ve tahmin yapılabilecek sürece yardımcı olabilecek bilgilere dönüştürmektedir. Tahmin modellerinin performansını iyileştirebilecek olan önemli özelliklerin doğru kombinasyonunu seçmek için kullanılmaktadır (Shouman et al., 2013).

Veri madenciliğinin tanımı ham verilerden anlamlı bilgilere geçiş sürecidir. Ham veriler arasından gizli olan bilgileri çıkarmak için veriler çeşitli veri madenciliği yöntemleri uygulanarak işlenmektedir. Veri madenciliğinin bir sonraki aşamasında işlenmiş olan veriler kullanılabilir hale gelmektedir. Veri madenciliği ve veri işlemede kullanılmak üzere birçok açık kaynak ve ticari uygulama bulunmaktadır (Michael and Gordon, 2000).

Veri madenciliğinin çeşitli tanımları literatürde yer almaktadır. Büyük miktardaki veriler içerisinden “değerli” olan bilgiyi elde etme süreci olarak adlandırılabilmektedir (Yalçın, 2013). Benzer bir ifade ile veri tabanlarında bilgi keşfi, daha önce hiç keşfedilmemiş, potansiyel olarak yararlı ve anlaşılır olan bilgilerin büyük miktarlardaki verilerden çıkarılması sürecine veri madenciliği yardımcı olmaktadır. Veri madenciliği olarak değerlendirilen, Backend veritabanı yönetim sistemleri, istatistikler, yapay zeka, makine öğrenmesi, paralel ve dağıtık süreçler gibi veri analiz teknikleri bulunmaktadır (Han et al., 2022).

Gelecekle ilgili tahminler yapmamızı sağlayan bilgisayar programlarını kullanan veri madenciliği yöntemi; büyük miktardaki verilerden ilişki ve kuralların aranmasını sağlamaktadır. Geleceğin, en azından yakın geleceğin geçmişten çok farklı olmayacağı düşünülürse, geçmişte çizilen kurallar gelecekte de geçerli olacak ve gelecek için doğru tahminler yapmamızı sağlayacaktır. Veri tabanlarında bilgi keşfinin ilk adımları 1990'lı yıllarda atılmış ve veri madenciliği yeni teknolojiler ile birlikte yaygın olarak kullanılan yeni bir standart haline getirilmiştir (Alpaydın, 2000). Bilginin keşfi için belirli bir süreç gerekmektedir, süreci tamamlamak için ham veriyi kullanılabilir hale getirmek ve işlemek gerekmektedir (Kayaalp ve Başarslan, 2018).

****

**Şekil 2.2.1: Veri Madenciliği ve İş Zekası** (Cifci ve Hussain, 2018).

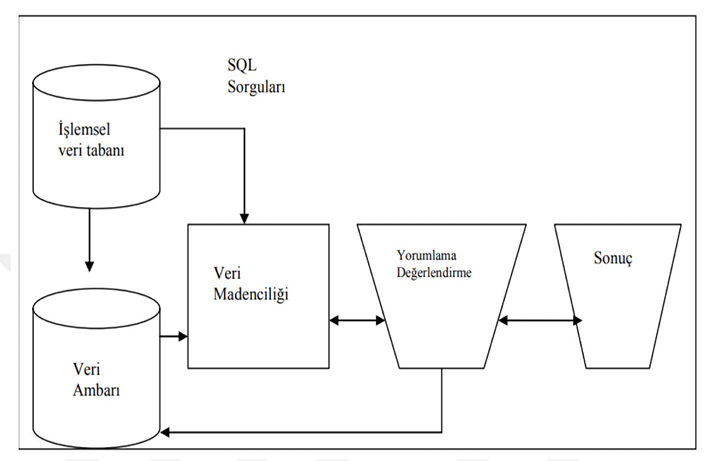
***6.1.1. Veri madenciliğinin tanımı***

Veri tabanlarında Öz Bilgi Keşfi (Knowledge Discovery in Databases) adı altında, sürekli ve yeni arayışlara yöneltilmesinin nedeni geleneksel sorgu ve raporlama araçlarının veri yığınları karşısında yetersiz kalması olarak belirtilmektedir. Veri tabanlarında öz bilgi keşfi süreci içerisinde, model kurulması ve değerlendirilmesi aşamalarında ortaya çıkan veri madenciliği (data mining) en önemli bölümü oluşturmaktadır (Akpınar, 2000).

Veri madenciliğin Veri Tabanlarında Öz Bilgi Keşfi’nin temelini oluşturan keşif kısmının gerçekleştiği adım olarak değerlendirilebileceği gibi bağımsız süreç olarak da değerlendirilmektedir.

***6.1.2. Veri madenciliği süreci ve aşamaları***

Şekil 2.2.2.1’de veri madenciliği süreci görsel olarak özetlenmektedir. Şekilde işlemsel veri tabanı, veri ambarı, veri madenciliği, yorumlama değerlendirme ve sonuç arasında çeşitli bağları gösteren bir görsel yer almaktadır (Akpınar, 2000). Farklı bilgi sistemlerinde dağınık bulunan verilerden önemli bilgileri alabilmek için veri ambarı 6 oluşturmak kullanılan bir tekniktir. Veri analizi ve veri madenciliği tekniklerinin kullanılmasının bir nedeni ise veri ambarındaki bilgilerin daha iyi kullanılabilmesi içindir (Kayaalp ve Başarslan, 2018).



**Şekil 2.2.2.1: Veri Madenciliği Süreci** (Oğuzlar, 2004).

1. **YÜKSEK BAŞARIMLI HESAPLAMA**

Yüksek başarımlı hesaplama, hesaplamalı işlemlerde zamanı kısaltan verimi artıran yazılım ve donanım içerikli faaliyetler bütünü olarak tanımlanabilir.

Günümüzde işlem hızını ve kapasitesini artırmaya yönelik olarak birçok yöntem ortaya çıkıştır. Bunlar arasında;

• Bellek Paylaşımlı Paralel Hesaplama

• Dağıtık Paralel Hesaplama

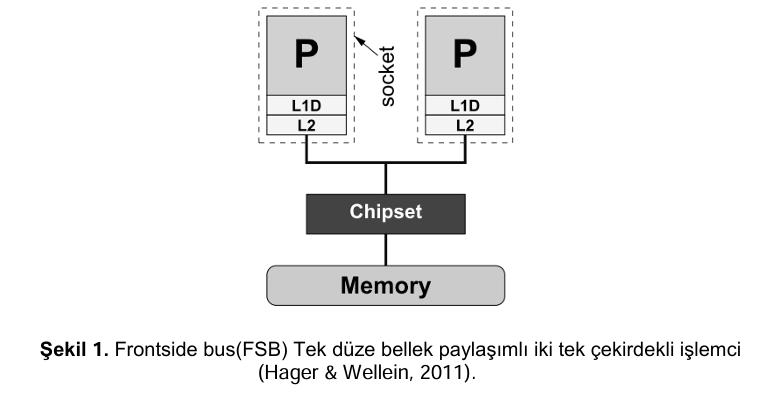
• Grid

• GPU (Grafik işlem birimi )kullanarak hesaplama

Paralel Hesaplamalı sistemleri paylaşımlı ve dağıtık olmak üzere iki sınıfa ayırabiliriz. Hesaplama yöntemleri önemli bir yere sahiptir. Bu yöntemlerin hepsine bulut bilişim adı verilmektedir. Gelişim açısından bulutlar (platform, altyapı, yazılım) bilgi teknolojilerinin gelişimindeki üçüncü dalga olarak görülmektedir (Udoh, 2011).

Paralel programlar, yoğun hesaplama sorunlarını çözmek için özel olarak tasarlanmıştır (Lea, 1999). Görüntü işleme, örüntü tanıma, gibi birçok veri madenciliği ve yapay zeka uygulamalarında zamanı kısaltarak daha çok algoritma testi yapmaya imkan vermekte ve dolayısıyla performansı artırıcı bir rol oynamaktadır. Yöntemi ne olursa olsun sonuçta hız ve dolayısı ile zamandan tasarruf ana önceliktir. Ancak unutulmaması gereken bir diğer noktada veri madenciliği araştırmalarında bir biri ile ilişkili verileri bulmak üzere yapılan analizlerde büyük veriler karşımıza çıkmaktadır. Büyük veriler ise donanım yetersizlikleri ile zaman kaybını doğurabilmektedir.

***7.1 Bellek Paylaşımlı Paralel Hesaplama***

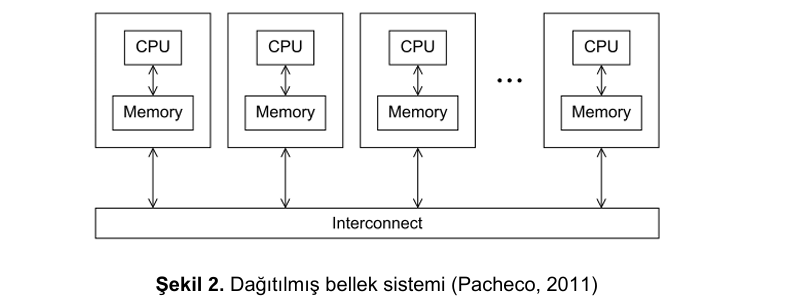
Bir problemin çözümü için çok sayıda hesaplama elemanı (core-çekirdek) bir araya getirildiğinde paralel hesaplamadan söz ediyoruz demektir. Tüm modern süper bilgisayar mimarileri ağırlıkla paralel bağlıdır ve büyük ölçekli süper bilgisayar işlemci sayısı giderek artmaktadır (Hager & Wellein, 2011). Bellek paylaşımlı paralel bilgisayarlar, paylaşılan fiziksel adres alanında ortak çalışan mikroişlemciler sistemidir (Hager & Wellein, 2011)

Paylaşılan bellekli sistemlerde, şekil 1.’de görüldüğü gibi her işlemci veya her çekirdek doğrudan belleğe erişebilir. Dağıtılmış bellek sistemlerinde ise her işlemcinin kendine özgü belleği vardır. Bazı sistemlerde ise nispeten küçük özel bellek paylaşılan bellek olmak üzere hibrid bir yapı kullanılır (Pacheco, 2011).

Bellek paylaşımı tabanlı sistemlerde aynı olan çok sayıda işlemci uyumlu saat frekansı ile işleme başlar ve aynı belleği paylaşırlar (Çelik & Özmen, 2009). İşlemci üzerinde paralellik, bir görevin bölümlerinin işlemci grubu arasında eşit dağıtılması yoluyla gerçekleştirilir. Öncelikle belirlenen kurala ve dağıtılmış veriye göre her işlemci kendi hesaplamasını yapar ve sonucu önceden belirlenmiş olan işlemciye gönderir. Bu süreç yazılım yoluyla gerçekleştirildiğinden esnek ve ekonomik bir yöntemdir (Çelik & Özmen, 2009).

***7.2 Dağıtık Paralel Hesaplama***

Bir programcı bakış açısından dağıtılmış bellekli bir sistem, şekil 2.’de görüldüğü gibi çekirdek koleksiyonunun bellek çiftleri halinde bir ağa dağıtılmasıdır.



1990 ların başlarında dağıtık hesaplama da ise temel sorun parçalar arasındaki iletişimin kurulmasıydı. Bu sorunun çözümü için geliştirilen mesaj geçiş arayüzü MPI (Message Passing Interface) günümüzde ticari veya ücretsiz uygulamalarda kullanılan standart haline geldi (Hager & Wellein, 2011).

Günümüzde MPI dağıtık sistemler arasında standart iletişim yöntemi haline gelmiştir. Her işlemci, yerel belleğine doğrudan, diğer belleklere ise oradaki işlemcilere ileti yollayarak erişir (Eraslan, 2007). Doğrudan veya dolaylı fiziksel ara bağlantı ağları ile işlemciler arasında mesaj gönderilip alınabilir. Eğer iki işlemci doğrudan birbirine bağlı değilse ara düğümler yardımıyla (MPI) bu bağlantı gerçekleştirilir. MPI, işlemciler arası iletişim, veri koordinasyonu ve senkronizasyonu gibi görevleri ara bağlantıyı kullanarak gerçekleştirir (Rauber & Rünger, 2010).

Bu yöntemin avantajı işlemci sayısı ile doğru orantılı olarak belleğinde artmasıdır. Bellek erişimi ağ ortamına girilmediği için daha hızlı olur. İşlemciler arasında iletişim sorunları yaşanması, yük dağılımının programcı tarafından dengelenmesi bu hesaplama yönteminde dezavantaj oluşturan bir durumdur.

***7.3 Grid Hesaplama***

Distributed computing olarak adlandırılan dağıtık bilgi işleme yönteminin sanallaştırılmasını sağlayan çözüm mimarisine kısaca Grid Hesaplama denilmektedir.

Grid hesaplamadaki temel amaç dağıtık bilgi işleme ve veri kaynaklarının kullanmakta olduğu işlemci güçleri, ağ kapasiteleri ve depolama kapasiteleri ile tek büyük bir sistem yaratmaktır ve oluşturulan bu sistem tamamen birbirinden bağımsız çalışmakta olan ve birbirine benzemeyen sistemlerin bir araya gelerek oluşturduğu sanal bir işleme gücüdür (IBM).

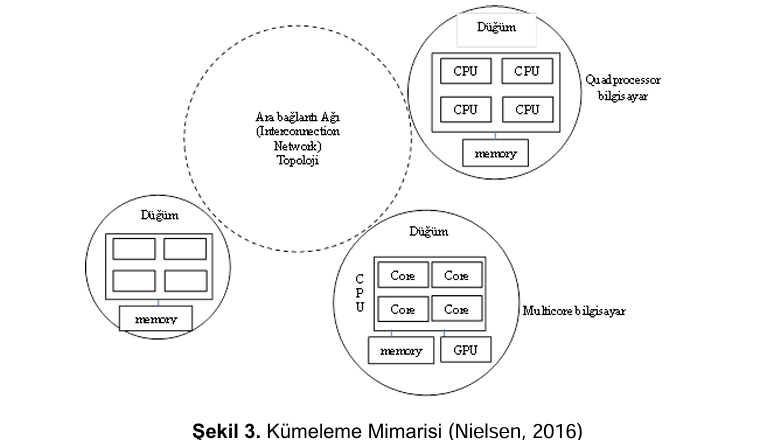
Küçük tesislerin elektrik ihtiyaçlarını tüm hat üzerinden karşılamaları grid yapısı için bir örnek olabilir. Grid yapısı yüksek güç sağlayan büyük ölçekli tedarikçiler olarak görülebilir (Resch & Gabriel, 2011).

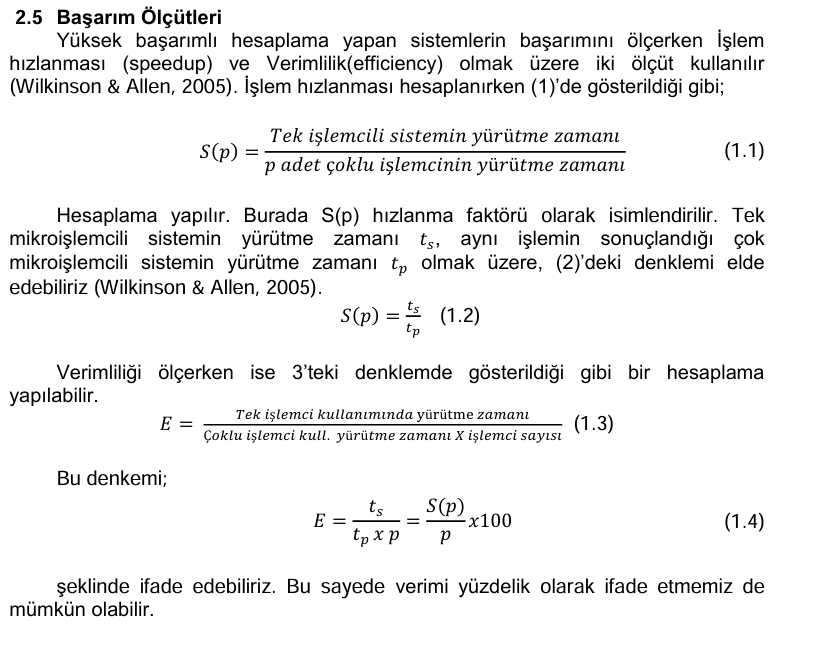
***7.4 Dağıtık Hesaplama***

Küme hesaplamanın temel felsefesi, şekil 3’de gösterildiği gibi ayrık durumdaki CPU ve RAM gibi hesaplamayı gerçekleştiren temel bilgisayar kaynaklarının düğümler halinde bir görev için organize edilmesidir. Dağıtık hesaplama ise birden çok bilgisayarda toplu olarak ortak bir amacı gerçekleştirmek için CPU birlikteliği kuran yöntemdir (Vasoya & Koli, 2016).

Dağıtık ve küme hesaplama, büyük veri kümelerini basit programlama modelleri kullanarak bilgisayar kümleri üzerinde işlemektir (Yang, Zhang, Hu, & Lin, 2015). Dağıtık hesaplamanın değişik ve zorlu görevleri vardır (Sinha, Saini, & Srikanth, 2014).

Bunlar arasında, iş parçalarının tanımlanması, bu parçaların eş zamanlı ve paralel olarak çoklu işlemcilere eşlenmesi, giriş çıkışları dağıtma ve programla ilişkilendirme, çoklu işlemciler ile veri paylaşımının yönetimi ve senkronize edilmesi sayılabilir (Vasoya & Koli, 2016). Küçük ve orta ölçekli bir ağ mimarisi için, bir veya daha fazla switch kullanılarak cluster düğümleri birbirine bağlanabilir. Bu bağlantıya ek te yapılabilir ve alt yapı olarak 1 veya 10 GBps’lik kabin bağlantıları kurulabilir (Guo, 2013). Bu çalışmada kümeleme yöntemi kullanılmıştır.

Ağa bağlanan her bilgisayar gerekli konfigürasyonlar yapıldıktan sonrakümenin bir parçası olur. Küme içerisinde genellikle donanım ve iletişim açısından en avantajlı 5 6 olan bilgisayar ana düğüm olarak belirlenir. Diğer bilgisayarlar ise hesaplama düğümü olarak mimaride yerini alır. Ana düğüm noktasında literatürde iş işleyicisi (JobTracker) ve iş yöneticisi (JobManager) olmak üzere genellikle iki bileşen vardır. Bu bileşenler düğümler arasındaki iş dağılımını ve yönetimini sağlar. Kümeye giren her bilgisayar aynı özelliklere sahip olmayabilir. Ortak ağ protokolünü desteklemeleri yeterlidir ki günümüzde çoğu sistem bunu destekler.



**8. YAZILIM ÖRNEKLERİ**

Günümüzde yüksek başarımlı hesaplamayı destekleyen yazılımların sayısı çok fazla olmamasına karşın, bu alandaki önemin ve ihtiyacın artmasıyla birlikte giderek sayıları artmaktadır. Bu yazılımları işletim sistemleri ve paket programlar olarak iki gruba ayırabiliriz.

**8.1 İşletim Sistemleri**

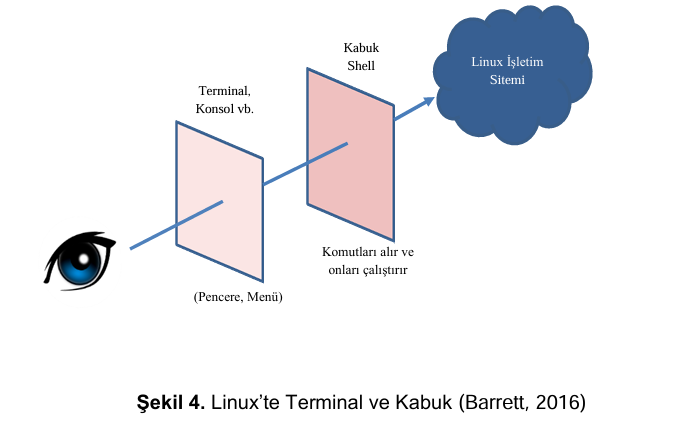
İşletim sistemlerinde yüksek başarımlı hesaplamanın verimini artırmak için gerekli olan bileşenlere baktığımızda, özellikle MPI (Message Passing Interface) kullanımı ve cluster desteği son derece önemlidir. Yüksek başarımlı hesaplama için en yaygın kullanılan işletim sistemleri genellikle Linux çekirdeğine sahip olan Red Hat Fedora, SUSE, ve CentOS dağıtımlarıdır (Eadline, 2009).

Microsoft’un server ailesi yazılımlarına ise sonradan eklenen HPC Pack isimli eklenti ile cluster oluşturma, yönetme ve izleme gibi yüksek başarımlı hesaplama sistemlerinde olması gereken birçok özellik yüklenebilmektedir. Microsoft HPC Pack teknolojisi, Windows Server üzerine inşa edilmiş bir yüksek performanslı hesaplama (HPC) çözümüdür. Windows HPC çözümü, Windows HPC küme ortamı için dağıtım, yönetim, iş planlaması ve izleme araçları ile HPC uygulamaları geliştirmek ve çalıştırmak için esnek bir platform sunar (Microsoft, 2014).

Windows ve Linux sık kullanılan işletim sistemlerindendir. Kullanıcılarına benzer işlemler sunsalar da çalışma mekanizmaları olarak Linux ile Windows arasında büyük farklar vardır. İşletim sistemi üzerinde çalışan uygulamalar, sistem kaynaklarına erişmek için sistem çağrıları yapar. Windows ve Linux'te sistem çağrıları davranış olarak benzer görünseler de, farklı bileşen ve fonksiyon tanımları kullanırlar.

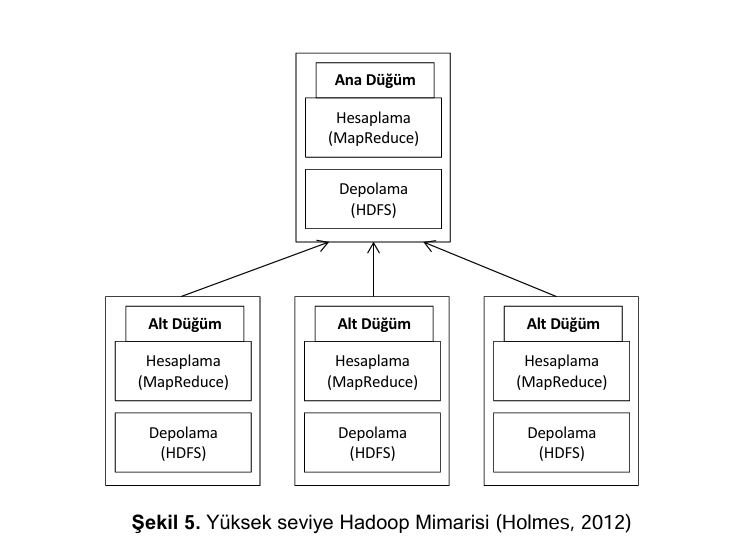
Windows sistem çağrılarını DLL mekanizması üzerinden API (Application Programming Interface - Uygulama Programlama Arayüzü) ile alır (Li, Yang, & Ma, 2012). Linux ve Windows arasındaki en belirgin fark, çekirdek yapıları ve dosya sistemlerinde kendini gösterir (Skendzic, Kovacic, & Jugo, 2011).

Windows işletim sistemi, merkezi uygulama tabanlı program bölümlerinden oluşur. Çekirdek alanında, çalışma alanı, çekirdek, aygıt sürücüleri ve donanım katmanı süreçleri yürütülür. Çekirdek bölümündeki programlar sistem verilerine ve donanımlara erişebilirler. Geliştirilen yazılımlar kullanıcı modunda çalışır ve sistem verileri ile donanımlara erişimleri sınırlıdır. Windows, yüksek modüler mimariye sahiptir. Her sistem fonksiyonu işletim sisteminin bir bileşenini kontrol eder. Tüm yazılımlar, sistem fonksiyonlarına erişmek için standart arayüz olan API’yi kullanır (Stallings, 2012).



Linux işletim sistemi açık kaynak kodlu UNIX tabanlı bir işletim sistemidir. Diğer popüler işletim sistemleri gibi simgeler, pencereler ve fare kontrolü ile grafik tabanlı kullanıcı ara yüzüne sahiptir. Linux’ün gerçek gücü, komutların yazılıp çalıştırıldığı komut satırı arayüzü(terminal) ve Shell(kabuk) çağrılarıdır. Şekil 1’de gösterildiği gibi terminal ve kabuk birbirinden farklıdır. Terminal görsel ara yüzden kabuk ise komutları alma ve çalıştırma ile görevlidir. Birçok Linux komutu girdiyi alıp ve çıktıyı üretir. Linux dosya sistemi parçalıdır. Root dizini en üst seviyedir ve diğer dizin ve dosyalar kökten aşağı doğru yayılırlar (Barrett, 2016). İşletim sistemi, dosyalama, depolama, ağ bağlantısı ve diğer ihtiyaçları karşılar. Çoğu kullanıcı nadiren çekirdeğe dikkat eder (Barrett, 2016).Çekirdek, işletim sisteminin merkezinde bulunan bir bileşendir. Temel olarak donanım ve uygulama yazılımlarını birbirine bağlar. Bir çekirdek oluşturulurken birbirine bağlı iki ana mimari vardır: mikroçekirdek(çok küçük parçalar halinde) ve monolitik çekirdek (büyük benzersiz tek parça). Linux’ün birçok dağıtımı vardır ancak temelde çekirdekleri benzer yapıdadır (Castro, 2016).

***8.2 Geliştirme Ortamları ve Paket Programlar***

Yapay zeka ve veri madenciliği uygulamalarına baktığımızda önemli ölçüde bilgisayarların çeşitli algoritmalar için hesaplama amacıyla kullanıldığını görürüz. Bu nedenle hesaplama yazılımlarının birçoğu önemli bir sorun haline gelmiş olan hesaplama süresi ve sistem gereksinimlerini azaltmak için bünyelerinde yüksek başarımlın hesaplama yöntemlerini gerçekleştirmeyi sağlayan eklentiler ve ara yüzler geliştirmişlerdir. Bu ise paralel çalışmayı destekleyen programlar yazmaya imkan veren yazılım geliştirme ortamları sayesinde mümkün olmuştur. OpenMp Archtecture Review Board(OpenMP ARB), donanım ve yazılım üreticileri birliği tarafından kurulmuş bir birliktir.OpenMP, bellek paylaşımlı paralel programlamaya imkan veren, birçok işletim sistemi mimarisinde çalışabilen (Solaris, IBM AIX , HP-UX, GNU/LINUX, MAC OS X ), basit ve esnek bir arayüz sağlayan, taşınabilir, ölçeklenebilir, çok çekirdekli işlemcilerle uyumlu, API desteği olan bir uygulama geliştirme arayüzüdür (OpenMP ARB, 2015). Java ve. NET programlama dilleri paralel programlamaya imkan vermektedir. Bu amaçla paralel programlamayı desteklemek için Paralel FX kütüphanesi (PFX) ve Task Parallel Library (TPL)’i geliştirmişlerdir. Bu sayede paralel iş parçacıkları geliştirmek ve programlamak mümkün olabilmektedir (Sen, Ranjan;Microsoft Corporation, 2008).Veri madenciliğinde büyük veri başlı başlına bir uğraşı alanıdır. Büyük verilerin depolanması, hızlı ve verimli bir şekilde analiz edilmesi günümüzde önemli bir bariyerdir. Bu bağlamda Linux dağıtımları üzerinde kullanılabilen Apache Hadoop geliştirilmiştir. Hadoop, birbirine paralel çalışan yüzlerce, binlerce hesaplama düğümü ile çok büyük veri setlerinin işlenmesi için geliştirilmiş yüksek ölçekli, açık kaynak kodlu depolama platformudur (IBM International Business Machines Corp. ). Paralel veri işleme ve büyük veri türlerini işlemek için uygun bir platform sağlar. Hadoop’un HDFS (Hadoop Distributed File System) ve MapReduce/YARN olmak üzere iki ana bileşeni vardır. Şekil 5.’te görüldüğü gibi ana düğüm noktası üzerinde ve alt(ikincil) düğüm noktaları üzerinde de benzer yapı bulunur ve ana düğüm noktasına bağlıdılar. İkincil düğüm noktalarına daha fazla düğüm eklemek mümkündür (Holmes, 2012)

HDFS, Hadoop’un dağıtık dosya sistemi bileşenidir. Büyük dosyalar üzerine yazma ve okuma görevlerinde yüksek başarıma sahiptir. Veri kaybına karşı donanım hatalarını yazma hatalarını tolere edebilecek bir sisteme sahiptir (Holmes, 2012). MapReduce, büyük veri setleri oluşturmak ve işlemek için programlama modelinin uygulama bileşenidir. Özel bir eşleşme fonksiyonu, ortak anahtar ve değer çiftlerini belirleyip oluşturarak bunları ortak anahtar/değer çifti şeklinde işler. Bir indirgeme fonksiyonu tüm ortak değerleri benzer ortak anahtarlar ile birleştirir (Dean & Ghemawat, 2004). MapReduce programlama modeli birçok farklı amaç için Google tarafından başarıyla kullanılmıştır. Bu model paralel ve dağıtık sistemler konusunda uzman olmayan programcılar tarafından kullanımı kolaydır (Dean & Ghemawat, 2004). Hadoop, içerisinde bulunan HDFS ve MapReduce bileşenlerinin üzerinde birçok aracıda ihtiva eder. Bunlar arasında;

• Yüksek Seviye Diller: Crunch, Cascading, Pig, Hive.

• Tahminleme Araçları: RHadoop, RHIPE, R, Mahout sayılabilir.

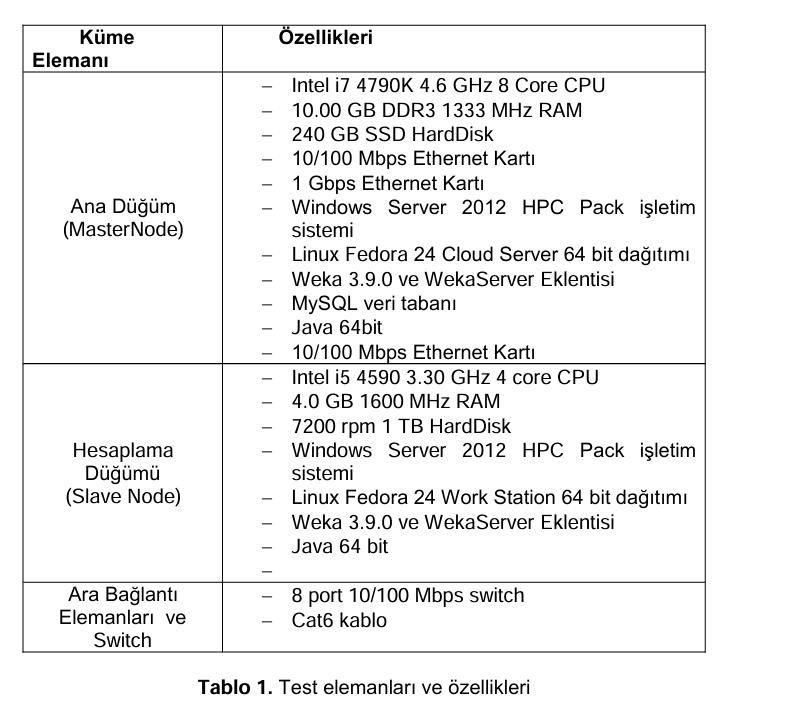
Bu araçlar yardımıyla istemciler Hadoop ile bağlantı kurarak tahminleme analizlerini gerçekleştirebilirler. Google, Facebook, Twitter, Yahoo!, eBay, Samsung, AOL ve daha birçok büyük şirket Hadoop kullanmaktadır (Holmes, 2012). Hadoop platformunun özellikle veri analizinde birçok analist tarafından tercih edilen R yazılımına uyumlu olması önemlidir. MathWorks şirketi tarafından geliştirilmiş olan Matlab yazılımı paralel hesaplama teknikleri ve Hadoop kullanımına uygun araçlar sunmuştur. Masaüstü bilgisayarlar için MapReduce kullanımını da sunmaktadır (The MathWorks, Inc.). Yine Stata yazılımı da paralel hesaplama desteği mevcut bir diğer veri analiz programıdır. Weka(Waikato Environment for Knowledge Analysis), Waikato Üniversitesinde geliştirilmiş, makine öğrenmesi alanında birçok algoritmayı ve aracı içerisinde 9 barındıran bir yazılımdır. Bu yazılıma Ekim 2011 tarihinde, Pentaho şirketi tarafından paralel hesaplamayı desteklemek üzere WekaServer adında bir eklenti yayınlamıştır (Pentaho A Hitachi Group Company). Bu eklenti kurulduktan sonra istenildiği kadar hesaplama bilgisayarı ana düğüm noktasına bağlanabilir. Ana düğüm noktası makine öğrenmesi algoritmalarından gelen iş görevlerini hesaplama düğümlerine göndererek daha hızlı ve yük dağılımlı olarak hesaplanmasını sağlayabilir. Hesaplama düğümlerinin durumu ise web tabanlı olarak anlık görülebilmektedir.

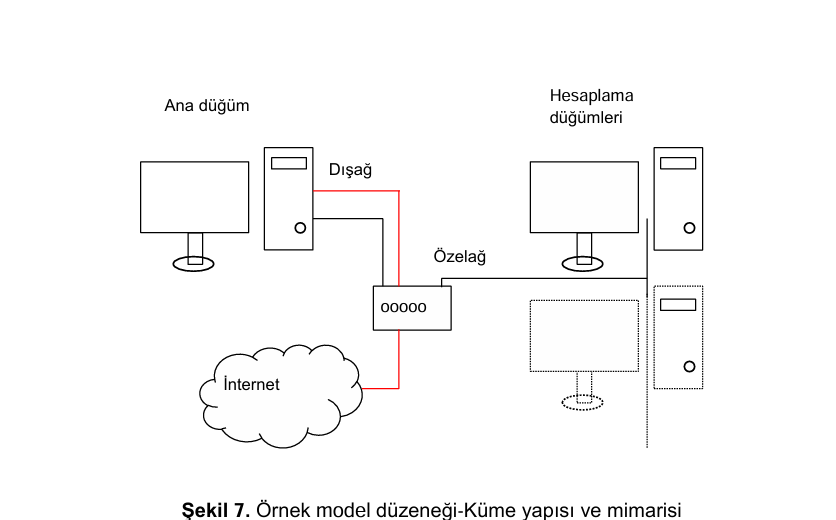
***8.3 Weka Yazılımı***

Bu çalışmada analiz ve deney yazılımı olarak Weka programı kullanılmıştır. Weka, Yeni Zellanda’daki Waikato Üniversitesi tarafından Waikato Environment for Knowledge Analysis adı ile geliştirilmiş yazılımdır. Sistem java programlama dili ile yazılmış ve GNU lisansı ile dağıtılmaktadır. Weka yazılımı birçok işletim sistemi altında çalışabilir. Platform bağımsız diyebiliriz. Tek bir ara yüzde çok sayıda öğrenme algoritmasını sağlar. Bunların yanında veri setleri üzerinde indirgenim gibi işlemleri yapmaya sağlayan düzenleme araçları da sağlar (Witten, Frank, & Hall, Data Mining Practical Machine Learning Tools and Techniques, 2011). JDBC sürücüsü sayesinde veri tabanı gibi kaynaklara erişim mümkündür. Kümeleme, sınıflandırma, regresyon birliktelik kuralları yöntemlerini kullanarak analizler yapabilir (Bell, Machine Learning Hands-On for Developers and Technical Professionals, 2015). Weka yazılımı için son zamanlarda geliştirilen Paket Yöneticisi (Package Manager) ile birçok yeni geliştirilmiş veya ek özellikler katılmış algoritma yanında WekaServer gibi kümeleme gerçekleştirmeye imkan sağlayan eklentiler ilave etmek mümkündür. Bu çalışmada kümeleme için WekaServer eklentisi kullanılmıştır.

**GEREÇ ve YÖNTEM**

Çalışmada kullanılan yazılımlar ve donanım mimarisi tablo1.’de gösterilmiştir. İki farklı işletim sistemi ana düğüm ve hesaplama düğümlerine yüklenmiş ve istenilen işletim sistemine göre başlangıç ekranında seçim yapılarak çalıştırılmıştır.

Tablo 1’de listelenen yazılım ve donanım elemanları şekil 7’de gösterildiği gibi kurulmuştur. Ana düğüm üzerine çift ethernet kartı bulunmaktadır. Bu sayede ana düğüm noktasında hesaplama düğümleri ve dış ağ arasındaki bağlantı aynı anda sağlanmıştır. Şekil 7’te görüldüğü gibi özel ağda bulunan hesaplama düğümlerine özel bir ip aralığı ile ip verilmiştir. Bu sayede dış ağ bloğu ile farklı ip bloğuna sahip olmuşlar ve dış ağ ile aralarındaki tek bağlantı ana düğüm olmuştur. Ana düğüm hesaplama düğümleri için ağ geçidi görevini de yerine getirmektedir. Hesaplama düğümlerinin sayısı istenildiği kadar artırılabilir. Bu çalışmada asgari gereklilik olan 1 ana düğüm ve 1 hesaplama düğümü bilgisayarı kullanılmıştır. Ana düğüm ayrıca etki alanı yöneticisi olarak kullanılmıştır.

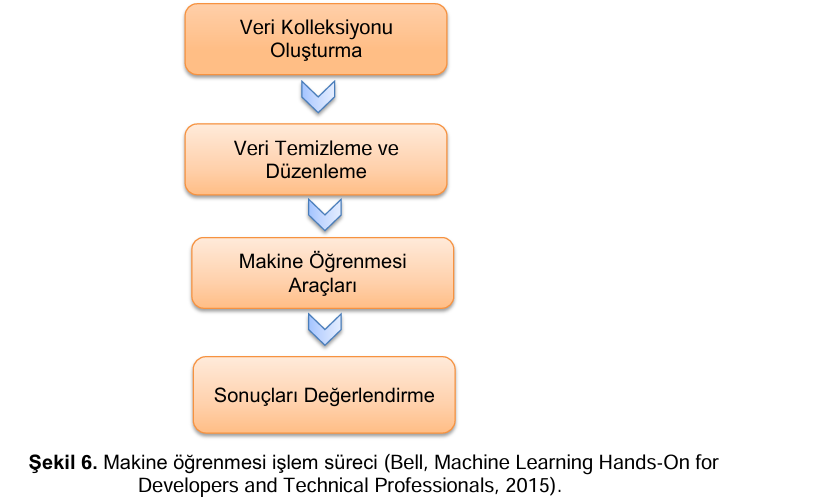


Ana düğüm noktası önce Windows işletim sistemi ile başlatılmış ve Weka yazılımı bu işletim sistemi üzerinde çalıştırılmıştır. Test için kullanılan veri seti Kaliforniya Üniversitesi Irvine kampüsü, Makine Öğrenmesi ve Uzman Sistemler Merkezi veri havuzundan alınan Skin Segmentation (Dhall & Bhatt, 2016) adlı veri setidir. Bu veri seti, 3.2 MB boyutunda, 245057 satır ve 4 sütundan oluşmaktadır. Text halinde indirilen bu veri seti ana düğüm de bulunan MySQL veritabanına aktarılmıştır. Ana düğüm noktası üzerinde Weka yazılımı MySQL veritabanına bağlanarak test verileri transfer edilmiştir. İlk olarak sadece ana düğümde Windows işletim sistemi çalıştırılarak, test verisi IBk(k-NN), MultiLayerPerceptron, Random Forest, Lineer Regresyon ve Decision Table makine öğrenmesi algoritmalarıyla ayrı ayrı işlenmiştir. Bu algoritmalar sık kullanılan makine öğrenmesi sınıflandırma algoritmalarıdır. Weka yazılımının bilgi ekranından her bir algoritmanın işleme başlama ve bitirme zamanları kaydedilmiştir. Aynı işlem ana düğümde Linux işletim sistemi çalıştırılarak tekrarlanmıştır. İkinci olarak algoritmalar ve test verisi, ana düğüm ve hesaplama düğümünde Windows işletim sistemi üzerinde WekaServer eklentisi yardımıyla küme hesaplama yöntemi ile analiz edilmiştir. Her bir algoritma için başlama ve bitiş zamanları yine kayıt altına alınmıştır. Son olarak ana düğüm ve hesaplama düğümü Linux işletim sistemi ile açılmış, WekaServer eklentisi kullanarak aynı işlemler tekrarlanmıştır.

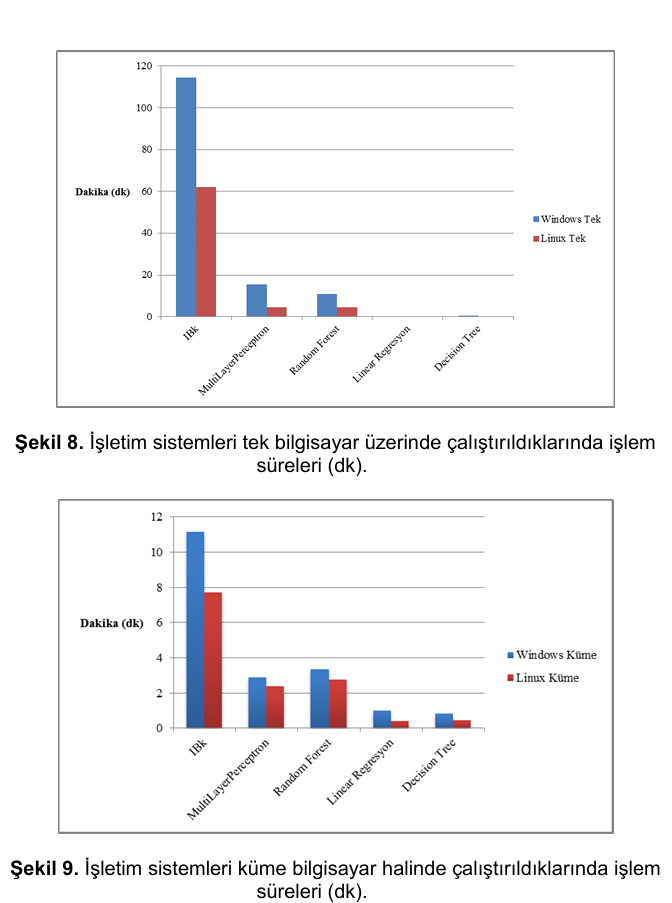
**8.4 Yapay Zeka ve Veri Madenciliği**

Makine öğrenmesi, bilgisayarların örnek veri veya geçmiş uzman deneyimlerini kullanarak, belirli ölçütlere göre başarımlarını artıracak biçimde programlanmasıdır (Alpaydın, 2013). Algoritması bilinmeyen durumlarda geçmiş deneyimler ve veriler kullanarak makine öğrenmesi yöntemleri girdinin çıktısını tahminleyebilir. Bunu geçekleştirmek için de öğrenme algoritmaları adı altında birçok algoritma geliştirilmiştir. Makine öğrenmesinin geçmişi 1950 yılına dayanır. 1950 yılında Alan Turing “Makineler düşünebilir mi?” sorusunu yöneltmiştir (Bell, Machine Learning Hands-On for Developers and Technical Professionals, 2015).

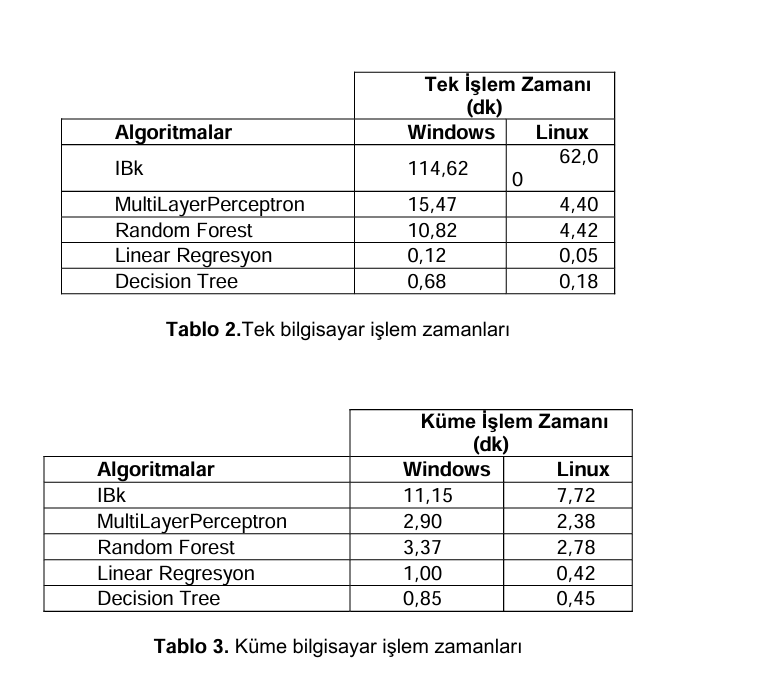
Günümüzde bu soruya hala cevap aranmakla birlikte önemli mesafeler alınmıştır. Geleneksel istatistik, veri kümeleri arasındaki ilişkileri tespit etmek için tümdengelim yöntemini kullanır, makine öğrenmesi, yapay sinir ağları gibi yapay zeka öğrenme teknikleri ise veri kümeleri arasındaki zayıf desenleri bulmak için endüktif yöntemi izler (Nisbet, Elder, & Miner, 2009). Makine öğrenmesi işlem süreci şekil 6.’da gösterilen adımlardan oluşur. Veri kolleksiyonu oluşturmak için günümüzde oldukça fazla sayıda ortam bulunmaktadır. Makine öğrenmesi sürecinin ilk iki aşaması veri madenciliği disiplininin ilgi alanına girer. Bu çalışma ise bu süreçte üçüncü basamakta gerçekleşen işlemlerle ilgilidir. Ortaya çıkan veri kümelerinin analizinde her araştırmacı yazılım ve donanım temelli bir düzenek kurmaktadır. Analizleri bu düzenek üzerinde gerçekleştirirken algoritmalardan faydalanır. Algoritma türlerinden biride makine öğrenmesi algoritmalarıdır. Günümüzde, yerel ağ veya kampüs türündeki ağlarda Windows ve Linux de dahil olmak üzere farklı işletim sistemlerinden oluşur ( Uemura, Nakajima, & Sato, 2007).

**BULGULAR ve TARTIŞMA/SONUÇ**

Bilgisayarda iki farklı işletim sistemi ile gerçekleştirilen analizlerde elde edilen işlem süreleri dakikaya çevrilerek şekil 8’de gösterilmiştir. Lineer regresyon ve Decision table algoritmaları çok kısa sürmüştür. Özellikle işlem yükü bakımından yoğun hesaplama gerektiren IBk ve MultiLayer Perceptron algoritmalarının çalışma sürelerine bakıldığında süreç daha iyi anlaşılmaktadır.



Küme halinde yine iki farklı işletim sistemi ile homojen oluşturulan düzenekte gerçekleştirilen deneyler sonunda elde edilen işlem süreleri ise şekil 9’de gösterilmiştir. Burada tek olarak çalışan bilgisayarların işlem süreleri ile küme halinde çalışan bilgisayarların işlem sürelerini karşılaştırırsak küme hesaplamanın süreyi ne kadar kısalttığı net görülebilir. Daha ayrıntılı veriler tablo 2 ve tablo 3’te gösterilmiştir. Tablolar incelendiğinde işlem süresi tek bilgisayarda çok kısa süren algoritma çalışma zamanlarının küme bilgisayar yönteminde arttığı görülmektedir.



Veri ekosisteminin artık oldukça geniş bir alana sahip olduğunu söyleyebiliriz. Bu ekosistemde doğal olarak ortaya çıkan devasa veri kümelerinin analiz edilerek veriden bilginin ortaya çıkarılması oldukça önemli ve bir o kadar da zor bir süreç haline dönüşmüştür. Büyük miktardaki verilerin analiz edilmesinde yüksek başarımlı hesaplama tekniklerini ve yazılımlarını kullanmak günümüz şartlarında oldukça avantajlı bir durum olarak karşımıza çıkmaktadır. Bu çalışmada bahsedilen yöntem ve yazılımlar araştırmacıların büyük veri analizi problemini çözmelerine yardımcı olabilir. En çok kullanılan yazılımlar, araçlar ve yetenekleri bu çalışmada araştırmacıların bilgisine sunulmaya çalışılmıştır. Özellikle Hadoop platformu büyük ölçekli şirketler tarafından sık kullanılmaktadır. Hadoop platformunu kullanmak isteyen araştırmacılar, özellikle Linux işletim sistemi kullanımı konusunda daha önceden deneyimleri yoksa sorun yaşayabilirler. Bunun yanında gelecekteki çalışmalarda yüksek başarımlı hesaplamaya imkan veren yazılımların test verileriyle birebir karşılaştırılması ve performanslarının görülmesi faydalı olabilir. Bu çalışmada elde edilen veriler analiz edildiğinde diğer bir ortaya çıkan sonuç küme hesaplama yönteminin tek bilgisayar yöntemine göre analiz sürelerini kısalttığıdır. Ancak tek bilgisayar üzerinde çok kısa süren analizlerin küme yöntemini gerektirmediği de görülmüştür. Çalışma da ortaya çıkan bir diğer sonuç ise önceki çalışmalarda sanal makine ortamında elde edilen sonuçların gerçek ortamda da görülmesi yani Linux işletim sisteminin küme hesaplama da işlem zamanı açısından daha avantajlı olduğudur. Windows ve Linux işletim sistemi küme hesaplama yöntemleri arasında çok büyük farklar olmasa da Linux’un çekirdek yapısından kaynaklana bir hız avantajının ve farkının olduğu da açıktır. Çalışmaya düzeneğin 15 kurulması açısından bakıldığında Linux işletim sistemi küme ayarlarının Windows işletim sistemine göre daha uzun sürdüğü görülmüştür. İki işletim sistemi arasındaki küme hesaplama açısından yapılan performans araştırması farklı kümeleme yöntemini destekleyen yazılımlarla tekrarlanarak benzer sonuçları verip vermeyeceği araştırmacılar tarafından incelenebilir. Bunun yanında bu iki sistemin RAM bellek kullanımları da araştırılabilir.

**9.YAPAY ZEKA VE VERİ MÜHENDİSLİĞİ UYGULAMALARI**

Yapay zeka ve veri mühendisliği, çeşitli alanlarda birçok uygulama alanına sahiptir. İşte yapay zeka ve veri mühendisliğinin yaygın uygulama alanlarından bazıları:

1. \*\*Sağlık Hizmetleri:\*\* Yapay zeka ve veri mühendisliği, hastalık teşhisi, tıbbi görüntüleme analizi, ilaç geliştirme süreçleri ve kişiselleştirilmiş tıp gibi sağlık hizmetlerinde birçok uygulamaya sahiptir. Örneğin, derin öğrenme yöntemleri, kanser teşhisi ve tedavisi için kullanılabilir.

2. \*\*Finansal Hizmetler:\*\* Finansal hizmetlerde, yapay zeka ve veri mühendisliği, dolandırıcılık tespiti, kredi riski değerlendirmesi, hisse senedi fiyat tahmini, müşteri ilişkileri yönetimi ve ticaret stratejilerinin otomatikleştirilmesi gibi birçok alanda kullanılır.

3. \*\*Perakende ve E-Ticaret:\*\* Yapay zeka ve veri mühendisliği, müşteri davranışlarının analizi, ürün önerileri, fiyatlandırma stratejileri, envanter yönetimi ve pazarlama kampanyalarının kişiselleştirilmesi gibi perakende ve e-ticaret sektörlerinde önemli rol oynar.

4. \*\*Ulaşım ve Lojistik:\*\* Ulaşım ve lojistik sektöründe, yapay zeka ve veri mühendisliği, rotalama optimizasyonu, teslimat süreçlerinin iyileştirilmesi, araç bakım tahminleri, trafik yönetimi ve otomatik sürüş gibi birçok alanda kullanılır.

5. \*\*Üretim ve Endüstriyel Otomasyon:\*\* Üretim ve endüstriyel otomasyonda, yapay zeka ve veri mühendisliği, işletme verimliliğini artırmak, arıza tahminleri yapmak, kalite kontrolü sağlamak ve üretim süreçlerini optimize etmek için kullanılır.

6. \*\*Enerji ve Doğal Kaynaklar:\*\* Enerji ve doğal kaynak sektörlerinde, yapay zeka ve veri mühendisliği, enerji tüketim tahminleri, akıllı grid yönetimi, enerji verimliliği analizi, çevresel etkilerin izlenmesi ve yenilenebilir enerji kaynaklarının entegrasyonu gibi birçok alanda kullanılır.

7. \*\*Eğitim ve Öğretim:\*\* Eğitim ve öğretimde, yapay zeka ve veri mühendisliği, öğrenci başarısı tahmini, öğrenme içeriğinin kişiselleştirilmesi, öğretim materyallerinin otomatik değerlendirilmesi ve öğrenci ilerlemesinin izlenmesi gibi birçok alanda kullanılır.

Bu uygulama alanlarının yanı sıra, yapay zeka ve veri mühendisliği, birçok başka sektörde de çeşitli uygulamalara sahiptir ve sürekli olarak yeni uygulama alanları keşfedilmektedir. Yapay zeka ve veri mühendisliğinin hızla gelişen teknolojileri ve çeşitli sektörlerdeki büyük veri miktarları, bu alanların önemini ve etkisini artırmaktadır.

***9.1.1* *sağlık sektöründe yapay zeka ve veri madenciliği***

Sağlık sektöründe yapay zeka ve veri mühendisliği, bir dizi önemli uygulama alanına sahiptir ve sağlık hizmetlerinin iyileştirilmesi, hastaların tedavisi ve sağlık uzmanlarının karar verme süreçlerinde önemli rol oynar. İşte sağlık sektöründe yapay zeka ve veri mühendisliğinin bazı temel uygulama alanları:

1. \*\*Hastalık Teşhisi ve Tahmini:\*\* Yapay zeka ve veri mühendisliği, hastalıkların erken teşhisi ve tahmini için kullanılabilir. Örneğin, derin öğrenme algoritmaları, tıbbi görüntüleme verilerini analiz ederek kanser gibi hastalıkları tespit edebilir veya kardiyovasküler riskleri değerlendirebilir.

2. \*\*Kişiselleştirilmiş Tıp:\*\* Yapay zeka ve veri mühendisliği, hastaların genetik verilerini, tıbbi geçmişlerini ve yaşam tarzı verilerini analiz ederek kişiselleştirilmiş tedavi ve önleme planları oluşturabilir. Bu, her hastanın benzersiz ihtiyaçlarına göre tedavi planlarının uyarlanmasını sağlar.

3. \*\*İlaç Geliştirme:\*\* Yapay zeka ve veri mühendisliği, ilaç keşfi, moleküler tasarım ve klinik denemeler gibi süreçlerde kullanılabilir. Makine öğrenimi algoritmaları, büyük veri setlerini analiz ederek potansiyel ilaç adaylarını belirleyebilir veya ilaç etkileşimlerini değerlendirebilir.

4. \*\*Hasta Bakımı ve Yönetimi:\*\* Sağlık sektöründe yapay zeka ve veri mühendisliği, hastaların bakımını optimize etmek ve yönetmek için kullanılabilir. Örneğin, hasta akışını tahmin eden modeller, hastane kaynaklarının daha etkili bir şekilde kullanılmasına yardımcı olabilir.

5. \*\*Doktor ve Hastane Performansının İyileştirilmesi:\*\* Yapay zeka ve veri mühendisliği, doktorların ve hastanelerin performansını değerlendirmek ve iyileştirmek için kullanılabilir. Bu, doktorların tedavi kararlarına rehberlik etmek, hasta sonuçlarını izlemek ve hasta memnuniyetini artırmak için kullanılabilir.

6. \*\*Tıbbi Görüntü Analizi:\*\* Yapay zeka, MR görüntüleri, röntgenler, histopatoloji görüntüleri ve diğer tıbbi görüntülerin analizinde kullanılabilir. Derin öğrenme algoritmaları, tıbbi görüntülerdeki anormallikleri tespit etmek veya hastalıkları teşhis etmek için kullanılabilir.

7. \*\*Epidemiyoloji ve Halk Sağlığı:\*\* Yapay zeka ve veri mühendisliği, hastalık salgınlarını izlemek, hastalık yayılımını tahmin etmek ve halk sağlığı politikalarını geliştirmek için kullanılabilir. Büyük veri analizi, hastalık salgınlarını erken tespit etmek ve yayılmasını kontrol altına almak için önemli bir araç olabilir.

Bu uygulama alanları, sağlık sektöründeki yapay zeka ve veri mühendisliğinin sadece birkaç örneğidir. Bu teknolojilerin sağlık hizmetlerine entegrasyonu, hastaların tedavi süreçlerini iyileştirmek, sağlık hizmetlerinin etkinliğini artırmak ve sağlık sonuçlarını optimize etmek için büyük potansiyele sahiptir.

***9.1.2 finans sektöründe uygulamaları***

Finans sektöründe yapay zeka ve veri mühendisliği, işlemlerin otomatikleştirilmesi, dolandırıcılık tespiti, kredi riski değerlendirmesi, portföy yönetimi, müşteri ilişkileri yönetimi ve hisse senedi fiyat tahmini gibi birçok alanda önemli uygulamalara sahiptir. İşte finans sektöründeki yapay zeka ve veri mühendisliğinin bazı temel uygulama alanları:

1. \*\*Dolandırıcılık Tespiti:\*\* Finansal kurumlar, yapay zeka ve veri mühendisliğini dolandırıcılık tespiti için kullanabilirler. Makine öğrenimi algoritmaları, kullanıcı hesaplarının ve işlemlerinin anomali ve dolandırıcılık belirtilerini tespit edebilir.

2. \*\*Kredi Riski Değerlendirmesi:\*\* Finansal kurumlar, kredi başvurularını değerlendirmek ve kredi riskini belirlemek için yapay zeka ve veri mühendisliğinden yararlanabilirler. Makine öğrenimi modelleri, müşteri verilerini ve finansal bilgileri analiz ederek kredi başvurularının riskini değerlendirebilir.

3. \*\*Portföy Yönetimi:\*\* Yatırım firmaları ve portföy yöneticileri, yapay zeka ve veri mühendisliğini portföy yönetimi süreçlerini optimize etmek için kullanabilirler. Makine öğrenimi algoritmaları, piyasa trendlerini analiz ederek portföy dağılımını optimize edebilir ve yatırım kararlarına rehberlik edebilir.

4. \*\*Müşteri İlişkileri Yönetimi:\*\* Finansal kurumlar, müşteri ilişkilerini güçlendirmek ve müşteri memnuniyetini artırmak için yapay zeka ve veri mühendisliğinden yararlanabilirler. Müşteri verilerini analiz eden makine öğrenimi modelleri, kişiselleştirilmiş hizmetler sunabilir ve müşteriye özel pazarlama kampanyaları geliştirebilir.

5. \*\*Hisse Senedi Fiyat Tahmini:\*\* Yatırım firmaları ve borsa yatırımcıları, hisse senedi fiyatlarını tahmin etmek için yapay zeka ve veri mühendisliğinden yararlanabilirler. Makine öğrenimi algoritmaları, piyasa verilerini ve teknik göstergeleri analiz ederek hisse senedi fiyatlarının gelecekteki hareketlerini tahmin edebilir.

6. \*\*Hızlı Ticaret ve Algoritmik İşlemler:\*\* Finansal piyasalarda hızlı ticaret ve algoritmik işlemler, yapay zeka ve veri mühendisliği tekniklerinden büyük ölçüde faydalanır. Makine öğrenimi algoritmaları, büyük miktarda veriyi gerçek zamanlı olarak analiz ederek karar alır ve algoritmik işlem stratejilerini otomatik olarak uygular.

7. \*\*Sigorta ve Risk Yönetimi:\*\* Sigorta şirketleri, risk değerlendirmesi, prim belirleme ve hasar tahminlemesi gibi alanlarda yapay zeka ve veri mühendisliğinden faydalanabilirler. Makine öğrenimi algoritmaları, sigorta risklerini değerlendirir ve sigorta primlerini optimize eder.

Kısa örnek;

Makine öğrenimi, sigorta ve risk yönetimi alanında önemli bir rol oynar. Örneğin, bir sigorta şirketi, müşterilerinin risk profillerini belirlemek ve prim fiyatlandırmasını optimize etmek için makine öğrenimi modellerini kullanabilir.

# Gerekli kütüphaneleri yükleme

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# Veri setini yükleme

data = pd.read\_csv("sigorta\_verisi.csv")

# Bağımsız değişkenler ve hedef değişkeni ayırma

X = data.drop("Sigorta\_Talebi", axis=1)

y = data["Sigorta\_Talebi"]

# Veri setini eğitim ve test setlerine ayırma

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Makine öğrenimi modelini oluşturma ve eğitme

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

# Modelin performansını değerlendirme

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("Model Doğruluğu:", accuracy)

bu kısa kod örneği

**10. YAPAY ZEKA VE VERİ MÜHENDİSLİĞİNDE ETİK**

Yapay zeka ve veri mühendisliği gibi alanlarda çalışırken, etik ilkelerin önemi büyüktür. Bu teknolojilerin güvenli, adil, şeffaf ve insan haklarına saygılı bir şekilde kullanılmasını sağlamak için belirli etik ilkelerin benimsenmesi gerekmektedir. İşte yapay zeka ve veri mühendisliğinde etik ilkelerin birkaç örneği:

1. \*\*Adillik ve Eşitlik:\*\* Yapay zeka ve veri mühendisliği çözümleri, tüm kullanıcılar için adil ve eşit şekilde çalışmalıdır. Algoritmalar, cinsiyet, ırk, etnik köken veya diğer demografik faktörlere dayalı ayrımcılık yapmamalıdır.

2. \*\*Gizlilik ve Güvenlik:\*\* Yapay zeka ve veri mühendisliği çözümleri, kullanıcıların kişisel verilerini korumalı ve gizliliğini sağlamalıdır. Veri güvenliği önlemleri alınmalı ve veriye erişim sıkı bir şekilde kontrol edilmelidir.

3. \*\*Şeffaflık ve Hesap Verebilirlik:\*\* Yapay zeka ve veri mühendisliği çözümleri, çalışma prensipleri ve algoritmaları hakkında şeffaf olmalıdır. Karar alma süreçleri ve sonuçları açık bir şekilde anlaşılabilir olmalıdır. Ayrıca, bu teknolojilerin kullanımıyla ilgili kararlar alırken hesap verebilirlik sağlanmalıdır.

4. \*\*İnsan Odaklılık:\*\* Yapay zeka ve veri mühendisliği çözümleri, insanların ihtiyaçlarına ve değerlerine saygı göstermelidir. Teknolojinin kullanımıyla ilgili etik ve sosyal etkiler dikkate alınmalı ve insan refahını artırmaya yönelik olmalıdır.

5. \*\*Toplumsal Fayda:\*\* Yapay zeka ve veri mühendisliği çözümleri, toplumun genel faydasına katkı sağlamalıdır. Teknolojinin kullanımıyla ilgili riskler ve fırsatlar dengelenmeli ve toplumun genel çıkarları gözetilmelidir.

6. \*\*Sorumluluk ve Risk Yönetimi:\*\* Yapay zeka ve veri mühendisliği çözümleri, olası risklerin belirlenmesi ve yönetilmesi için sorumluluk taşır. Geliştiriciler, kullanıcıları potansiyel riskler hakkında bilgilendirmeli ve riskleri en aza indirmek için gerekli önlemleri almalıdır.

Bu ilkeler, yapay zeka ve veri mühendisliği alanlarında çalışan profesyonellerin, geliştiricilerin ve karar vericilerin teknolojiyi etik bir şekilde tasarlaması ve kullanması için bir rehber sağlar. Bu ilkelerin benimsenmesi, teknolojinin insanlığa faydalı olmasını ve toplumsal değerlere uygun bir şekilde kullanılmasını sağlar.

***10.1.1 veri gizliliği ve güvenlik***

Veri gizliliği ve güvenliği, yapay zeka ve veri mühendisliği alanlarında büyük öneme sahiptir. Bu konular, kişisel verilerin korunması, hassas bilgilerin güvenliği ve kötü niyetli kullanımların önlenmesi gibi unsurları kapsar. İşte veri gizliliği ve güvenliği ile ilgili bazı temel kavramlar:

1. \*\*Kişisel Verilerin Korunması:\*\* Kişisel veriler, bireylerin tanımlanmasını mümkün kılan her türlü bilgiyi içerir. Bu verilerin toplanması, işlenmesi ve saklanması sırasında gizlilik ve güvenlik önlemleri alınmalıdır. Özellikle GDPR gibi düzenlemeler, kişisel verilerin korunmasını düzenler ve kurumları bu konuda sorumluluğa teşvik eder.

2. \*\*Veri Şifreleme:\*\* Veri şifreleme, verilerin yetkisiz erişime karşı korunması için kullanılan önemli bir güvenlik önlemidir. Hassas verilerin şifrelenerek saklanması, veri güvenliğini sağlar ve veri hırsızlığını önler.

3. \*\*Erişim Kontrolleri:\*\* Veri gizliliğini ve güvenliğini sağlamak için erişim kontrolleri önemlidir. Bu kontroller, verilere sadece yetkili kullanıcıların erişmesini sağlar ve yetkisiz erişimleri önler.

4. \*\*Veri Güvenliği Eğitimi:\*\* Kuruluş içindeki personelin, veri güvenliği konusunda eğitilmesi önemlidir. Bu eğitimler, veri güvenliği politikalarına uyumu artırır ve bilinçli bir çalışma ortamı oluşturur.

5. \*\*Veri Sızıntılarının Önlenmesi:\*\* Veri sızıntıları, kuruluşlar için ciddi bir risk oluşturur. Bu nedenle, veri sızıntılarını önlemek için güvenlik önlemleri alınmalı ve düzenli olarak güvenlik denetimleri yapılmalıdır.

6. \*\*Kötüye Kullanımı Önleme:\*\* Veri gizliliği ve güvenliği önlemleri, verilerin kötü niyetli kullanımını önlemeyi amaçlar. Bu, veri ihlallerini, veri hırsızlığını ve diğer kötüye kullanım vakalarını engellemeyi içerir.

Bu kavramlar, yapay zeka ve veri mühendisliği alanlarında çalışan profesyonellerin, kuruluşların ve bireylerin veri gizliliği ve güvenliği konularında dikkat etmesi gereken temel unsurları temsil eder. Bu ilkelerin uygulanması, veriye dayalı teknolojilerin güvenliğini sağlamak ve kullanıcıların güvenini kazanmak için önemlidir.

**11. GELECEK TRENDLER VE ÖNGÖRÜLER**

Gelecek trendler ve öngörüler, yapay zeka ve veri mühendisliği alanlarında hızla değişen ve gelişen teknolojileri ve endüstriyi açıklar. Bu trendler, teknoloji, iş dünyası, toplum ve küresel ekonomi üzerinde önemli etkilere sahip olabilir. İşte bu alanlarda önemli gelecek trendleri ve öngörülerden bazıları:

1. \*\*Genişleyen Yapay Zeka Uygulamaları:\*\* Yapay zeka, birçok endüstri ve sektörde giderek daha fazla uygulama bulmaktadır. Otomotiv, sağlık, perakende, finans, üretim ve daha birçok alanda yapay zeka teknolojilerinin kullanımı artmaktadır.

2. \*\*Daha İleri Derin Öğrenme Teknikleri:\*\* Derin öğrenme, yapay zeka alanında önemli bir gelişme alanı olmaya devam edecektir. Daha karmaşık yapay sinir ağı yapıları, daha büyük veri kümeleri ve daha iyi hesaplama kaynakları ile daha ileri derin öğrenme teknikleri geliştirilecektir.

3. \*\*Veri Tabanlı Karar Alma Süreçleri:\*\* Kurumlar, veriye dayalı karar alma süreçlerini iyileştirmek için daha fazla yapay zeka ve veri mühendisliği tekniklerine başvuracaklardır. Büyük veri analizi, tahminleme modelleri ve karar destek sistemleri, işletmelerin stratejik karar alma süreçlerine önemli katkılar sağlayacaktır.

4. \*\*Daha İyi Veri Güvenliği ve Gizliliği Çözümleri:\*\* Veri güvenliği ve gizliliği, yapay zeka ve veri mühendisliği alanlarında giderek daha fazla önem kazanmaktadır. Bu nedenle, daha güçlü şifreleme yöntemleri, daha iyi erişim kontrolleri ve daha etkili veri güvenlik çözümleri geliştirilecektir.

5. \*\*Yapay Zeka ve İş Süreçleri Otomasyonu:\*\* Yapay zeka ve robotik süreç otomasyonu, işletmelerin verimliliğini artırmak için kullanılan önemli bir teknoloji olmaya devam edecektir. Rutin iş süreçlerinin otomatikleştirilmesi, insan kaynaklarının daha stratejik görevlere odaklanmasını sağlayacaktır.

6. \*\*Daha İyi Kullanıcı Deneyimi:\*\* Yapay zeka ve veri mühendisliği teknikleri, daha iyi kullanıcı deneyimi sağlamak için kullanılacaktır. Kişiselleştirilmiş öneriler, hızlı müşteri hizmetleri ve akıllı kullanıcı arayüzleri, müşteri memnuniyetini artırmak için önemli bir rol oynayacaktır.

Bu trendler ve öngörüler, yapay zeka ve veri mühendisliği alanlarının gelecekteki gelişimine dair genel bir bakış sunar. Bu alanlarda çalışan profesyonellerin, bu trendleri takip etmek ve teknolojik gelişmeleri yakından izlemek önemlidir.

***11.1.1 gelecek fırsatlar ve zorluklar***

Gelecekteki yapay zeka ve veri mühendisliği alanında birçok fırsat ve zorluk öngörülmektedir. Bu alanların hızla gelişen teknolojileri ve endüstriyel uygulamaları, hem heyecan verici fırsatlar sunmakta hem de bazı önemli zorluklarla karşı karşıya getirmektedir. İşte bu alanlardaki gelecekteki fırsatlar ve zorluklardan bazıları:

\*\*Fırsatlar:\*\*

1. \*\*Yenilikçi Uygulamalar ve Endüstriyel Dönüşüm:\*\* Yapay zeka ve veri mühendisliği, birçok endüstriyel sektörde devrim niteliğinde değişimlere yol açmaktadır. Otomotiv, sağlık, finans, perakende, üretim ve daha birçok alanda yapay zeka ve veri mühendisliği uygulamaları, iş süreçlerini daha verimli hale getirecek ve yeni fırsatlar yaratacaktır.

2. \*\*Yenilikçi Ürün ve Hizmet Geliştirme:\*\* Yapay zeka ve veri mühendisliği teknolojileri, yenilikçi ürünler ve hizmetlerin geliştirilmesine olanak tanır. Kişiselleştirilmiş ürünler, akıllı cihazlar, özerk araçlar ve daha birçok yenilikçi çözüm, bu teknolojilerin kullanımıyla ortaya çıkabilir.

3. \*\*Sorunların Çözümü İçin Veri Odaklı Yaklaşımlar:\*\* Yapay zeka ve veri mühendisliği, birçok karmaşık sorunun çözümü için veri odaklı yaklaşımlar sunar. Sağlık sorunlarından çevresel problemlere, trafik yönetiminden enerji verimliliğine kadar birçok alanda veri odaklı çözümler geliştirilebilir.

4. \*\*Endüstriyel İnovasyon ve Rekabet Avantajı:\*\* Yapay zeka ve veri mühendisliği teknolojileri, şirketlerin rekabet avantajı elde etmesine ve endüstriyel inovasyonu teşvik etmesine olanak tanır. Veriye dayalı karar alma süreçleri, ürün geliştirme süreçleri ve işletme stratejileri, şirketlerin daha rekabetçi olmasını sağlayabilir.

\*\*Zorluklar:\*\*

1. \*\*Veri Gizliliği ve Güvenliği:\*\* Büyük veri setleri ve hassas bilgilerin kullanımıyla birlikte, veri gizliliği ve güvenliği büyük bir zorluk haline gelmektedir. Kişisel verilerin korunması, veri sızıntılarının önlenmesi ve siber saldırılara karşı korunma önemli bir endişe kaynağıdır.

2. \*\*Etik ve Hukuki Sorunlar:\*\* Yapay zeka ve veri mühendisliği teknolojilerinin kullanımıyla ilgili etik ve hukuki sorunlar ortaya çıkmaktadır. Algoritmik adalet, otomatik karar alma süreçleri ve yapay zeka sistemlerinin insan haklarına uygunluğu gibi konular, önemli tartışma ve araştırma alanlarıdır.

3. \*\*Kaynak Kısıtlamaları:\*\* Büyük veri kümelerinin işlenmesi ve derin öğrenme modellerinin eğitilmesi gibi işlemler, önemli hesaplama kaynakları gerektirir. Bu nedenle, kaynak kısıtlamaları ve altyapı zorlukları, yapay zeka ve veri mühendisliği çalışmalarını etkileyebilir.

4. \*\*Toplumsal Kabul ve Eğitim:\*\* Yapay zeka ve veri mühendisliği teknolojilerinin toplum tarafından kabul edilmesi ve anlaşılması önemlidir. Bu nedenle, teknolojiye yönelik toplumsal eğitim ve farkındalık önemli bir zorluk olabilir.

Bu fırsatlar ve zorluklar, yapay zeka ve veri mühendisliği alanında çalışan profesyonellerin, kuruluşların ve karar vericilerin dikkate alması gereken temel unsurları temsil eder. Bu alanlardaki gelişmelerin izlenmesi ve bu zorlukların üstesinden gelinmesi, teknolojinin toplum için maksimum fayda sağlamasını sağlar.

**SONUÇ VE ÖZET**

Yapay zeka ve veri mühendisliği, günümüzde teknolojinin en heyecan verici ve hızla gelişen alanları arasında yer almaktadır. Bu alanlar, bilgisayar bilimleri, istatistik, matematik ve mühendislik gibi disiplinlerin kesişiminde bulunur ve büyük veri setleri üzerinde analiz yaparak, örüntüleri belirleyen ve öğrenen algoritmalar geliştirirler. Bu algoritmalar, özellikle yapay sinir ağları, derin öğrenme ve makine öğrenimi gibi teknikleri içerir ve birçok farklı endüstriyel sektörde kullanılabilirler.

Yapay zeka ve veri mühendisliği alanları, birçok fırsat sunmaktadır. Örneğin, sağlık sektöründe yapay zeka, hastalık teşhisi ve tedavisi konusunda önemli bir rol oynayabilir. Finans sektöründe, yapay zeka ve veri mühendisliği teknikleri, risk analizi ve portföy yönetimi gibi alanlarda kullanılabilir. Perakende sektöründe, müşteri davranışlarını analiz etmek ve kişiselleştirilmiş pazarlama stratejileri geliştirmek için yapay zeka ve veri mühendisliği teknikleri kullanılabilir.

Ancak, bu alanların kullanımıyla ilgili bazı zorluklar da vardır. Özellikle, veri gizliliği ve güvenliği, etik sorunlar ve toplumsal kabul gibi konular, yapay zeka ve veri mühendisliği alanlarında çalışan profesyonellerin karşılaştığı önemli zorluklardır. Veri gizliliği ve güvenliği, büyük veri setlerinin işlenmesi sırasında hassas bilgilerin korunmasını gerektirirken, etik sorunlar, yapay zeka sistemlerinin insan haklarına uygunluğu ve adaleti sağlaması gerektiği konusunda önemli bir tartışma alanı oluşturur.

Sonuç olarak, yapay zeka ve veri mühendisliği alanları, geleceğin teknoloji ortamının önemli bir parçasını oluşturur. Bu alanlarda çalışan profesyonellerin, teknolojinin etik kullanımını teşvik etmeleri ve toplumun genel çıkarlarını gözetmeleri önemlidir. Gelecekte, yapay zeka ve veri mühendisliği teknolojilerinin daha da yaygınlaşması ve etkili bir şekilde kullanılmasıyla, birçok yeni fırsatın ortaya çıkması beklenmektedir. Ancak, bu teknolojilerin güvenliği, gizliliği ve etiği de göz önünde bulundurularak, dengeli bir şekilde kullanılması gerekmektedir.

**KAYNAKÇALAR**

Sözlü Bildiri: Özhan, E. (2016a). Farklı İşletim Sistemleri Üzerinde Kümeleme Yöntemi ile Makine Öğrenmesi Algoritmalarının Performans Testleri (Machine Learning Algorithms Performance Tests with Clustering Methods on Different Operating Systems.) In International Conference on Computer Science and Engineering (pp. 362–367).

Tekirdağ, Turkey: (UBMK 2016). Poster Bildiri: Özhan, E. (2016b). Yapay Zeka ve Veri Madenciliği Uygulamalarında Yüksek Başarımlı Hesaplama Yazılımları (High Performance Computing Software in Artificial Intelligence and Data Mining Applications). In International Conference on Computer Science and Engineering (pp. 719–724). Tekirdağ, Turkey: (UBMK 2016).

Uemura, Y., Nakajima, Y., & Sato, M. (2007). Direct Execution of Linux Binary on Windows for Grid RPC Workers. 2007 IEEE International Parallel and Distributed Processing Symposium (s. 1-8). Long Beach: IEEE. Akı, M., & Uçar, E. (2010). Trakya Üniversitesi Yüksek Başarımlı Hesaplama Sistemi (TRAKYAHPC) Kurulumu ve Web Arayüzü Tasarımı Üzerine Bir Örnek Olay . II. ULUSAL YÜKSEK BAŞARIMLI ve GRID HESAPLAMA KONFERANSI . İstanbul. Alpaydın, E. (2013). Yapay Öğrenme. İstanbul: MIT, BÜTEK. Aygün, S., & Akçay, M. (2015). Yüz Tanıma Teknolojilerinde Yüksek Başarım için Paralel Hesaplama. 4. Ulusal Yüksek Başarımlı Hesaplama Konferansı. Ankara. Barrett, D. J. (2016). Linux Pocket Guide. Sebastopol: O’Reilly Media, Inc. Bell, J. (2015). Machine Learning Hands-On for Developers and Technical Professionals. (C. Long, Dü.) Indianapolis, Canada: John Wiley & Sons, Inc. 16 Bell, J. (2015). Machine Learning Hands-On for Developers and Technical Professionals. Indianapolis, Indiana: John Wiley & Sons, Inc. Borriss, M., & Dannowski, U. (1998). TCP Performance over ATM on Linux and Windows NT. ICATM 98., 1998 1st IEEE International Conference on. Colmar. Castro, J. D. (2016). Introducing Linux Distros. New York : Springer Science+Business Media New York. Cormen, T. H., Leiserson, C. E., & Stein, C. (2009). Introduction to Algorithms. Cambridge, Massachusetts - London, England: The MIT Press. Çelik, A., & Özmen, A. (2009). Dağıtık Paralel Sistemler Hakkında Kıyaslamalı Bir Çalışma: PVM ve MPI. 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS’09). Karabük. Dean, J., & Ghemawat, S. (2004). MapReduce: Simplified Data Processing on Large Clusters. OSDI'04: Sixth Symposium on Operating System Design and Implementation. San Francisco. Dhall, A., & Bhatt, R. (2016, Temmuz). UC Irvine Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation adresinden alınmıştır Eadline, D. (2009). High Performance Computing For Dummies. Hoboken, NJ: Wiley Publishing, Inc. Eraslan, G. (2007, Mart 4). Paralel Programlama ve MPI. Temmuz 20, 2016 tarihinde https://seminer.linux.org.tr/wp-content/uploads/ParalelProgramlamaveMPI.pdf adresinden alındı Flynn, M. (1966). Very high-speed computing systems. Proceedings of the IEEE, 54(12), 1901-1909. Guo, S. (2013). Hadoop Operations and Cluster Management Cookbook. Birmingham, İngiltere: Packt Publishing Ltd. Hager, G., & Wellein, G. (2011). Introduction to High Performance Computing for Scientists and Engineers. Boca Raton: CRC Press Taylor & Francis Group. Holmes, A. (2012). Hadoop in Practice. New York: Manning Publications Co. IBM. (tarih yok). http://www-05.ibm.com/tr/solutions/edu/grid.html. (IBM) Temmuz 20, 2016 tarihinde http://www-05.ibm.com/tr/solutions/edu/grid.html adresinden alındı IBM International Business Machines Corp. . (tarih yok). Hadoop-IBM. (International Business Machines Corp. ) Temmuz 22, 2016 tarihinde http://www.ibm.com/analytics/us/en/technology/hadoop/#hadoop-resources adresinden alındı Lancaster, D., & Takeda, K. (1999). Comparative Performance of a Commodity Alpha Cluster running Linux and Windows NT. Cluster Computing, 1999. Proceedings. 1st IEEE Computer Society International Workshop on. Melbourne. Lea, D. (1999). Concurrent Programming in Java: Design Principles and Patterns,. Boston: Addison Wesley. 17 Li, R., Yang, N., & Ma, S. (2012). An Approach of Windows Memory Management Simulation on Linux. 2012 Third World Congress on Software Engineering. Wuhan. Microsoft;. (2014, Kasım 18). Microsoft Corp. (Microsoft) Temmuz 21, 2016 tarihinde https://technet.microsoft.com/tr-tr/library/cc514029(v=ws.11).aspx adresinden alındı Nielsen, F. (2016). Introduction to HPC with MPI for Data Science. (I. Mackie, Dü.) Switzerland, Switzerland: Springer International Publishing. Nisbet, R., Elder, J., & Miner, G. (2009). Handbook of Statistical Analysis and Data Mining Applications. Burlington, MA: Elsevier Inc. OpenMP ARB. (2015, Temmuz 14). The OpenMP® API specification for parallel programming. (The OpenMP® API specification for parallel programming) Temmuz 21, 2016 tarihinde http://openmp.org/wp/about-openmp/ adresinden alındı Pacheco, P. (2011). An Introduction to Parallel Programming. Burlington: Morgan Kaufmann Publishers is an imprint of Elsevier. Panda, M., & Nag, A. (2015). Plain Text Encryption Using AES, DES and SALSA20 by Java Based Bouncy Castle API on Windows and Linux. Second International Conference on Advances in Computing and Communication Engineering. Dehradun. Pentaho A Hitachi Group Company. (tarih yok). Weka Server. (Pentaho A Hitachi Group Company) Temmuz 24, 2016 tarihinde http://wiki.pentaho.com/display/DATAMINING/Weka+Server adresinden alındı Prajapati , H., & Vij , S. (2011). Analytical Study of Parallel and Distributed Image Processing. International Conference on Image Information Processing (ICIIP 2011). Himachal Pradesh. Rauber, T., & Rünger, G. (2010). Parallel Programming for Multicore and Cluster Systems. New York: Springer-Verlag Berlin Heidelberg. Resch, M., & Gabriel, E. (2011). Supercomputers in Grids. Cloud, Grid and High Performance Computing: Emerging Applications (s. 1-9). içinde Almanya: IGI Global. Ristov, S., & Gusev, M. (2013). Performance vs Cost for Windows and Linux Performance vs Cost for Windows and Linux. Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on. San Francisco. Sen, Ranjan;Microsoft Corporation. (2008, Eylül). Developing Parallel Programs. (Microsoft Corporation) Temmuz 21, 2016 tarihinde https://msdn.microsoft.com/en us/library/cc983823.aspx adresinden alındı Sinha, A., Saini, T., & Srikanth, S. V. (2014). Distributed Computing Approach to Optimize Road Traffic Simulation. IEEE Parallel, Distributed and Grid Computing (PDGC), (s. 360-364). Solan. 18 Skendzic, A., Kovacic, B., & Jugo, I. (2011). Decreasing Information Technology expenses by using emulators on Windows and Linux Platforms. MIPRO, 2011 Proceedings of the 34th International Convention. Opatija. Stallings, W. (2012). Operating Systems: Internals and Design Principles. New Jersey: Prentice Hall. Tanaka, K., Uehara, M., & Mori, H. (2008). A Case Study of a Linux Grid on Windows using Virtual Machines. 22nd International Conference on Advanced Information Networking and Applications. Okinawa. The MathWorks, Inc. (tarih yok). MATLAB MapReduce and Hadoop. (The MathWorks, Inc.) Temmuz 23, 2016 tarihinde http://www.mathworks.com/discovery/matlab-mapreduce hadoop.html?requestedDomain=www.mathworks.com adresinden alındı Udoh, E. (2011). Cloud, Grid and High Performance Computing: Emerging Applications. Fort Wayne Indiana: IGI Global. Vasoya, A., & Koli, N. (2016, Mart). Mining of association rules on large database using distributed and parallel computing. Procedia Computer Science, 79, 221-230. Werner, D. (2008). Data Mining Techniques in Grid Computing Environments. UK: John Wiley & Sons, Ltd. Wilkinson, B., & Allen, M. (2005). Parallel Programming Techniques And Applications Using Networked Workstations And Parallel Computers. Upper Saddle River, NJ: Pearson Prentice Hall . Witten, I. H., & Frank, E. (2005). Data Mining Practical Machine Learning Tools and Techniques (2 b.). San Francisco, CA, USA: Elsevier Inc. Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining Practical Machine Learning Tools and Techniques. Burlington,, Amerika Birleşik Devletleri: Morgan Kaufmann Publishers. Yang, Z., Zhang, C., Hu, M., & Lin, F. (2015). OPC:A Distributed Computing and Memory Computing based Effective Solution of Big Data. 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), (s. 50-53). Chengdu.

Chatgpt.com