YÜKSEK RAFLI DEPOLAMA SİSTEMLERİNİN ENERJİ OPTİMİZASYONUNDA ANOMALİ TESPİTİ İÇİN SINIFLAMA ALGORİTMALARININ KARŞILAŞTIRILMASI

***

COMPARISON OF CLASSIFICATION ALGORITHMS FOR ANOMALY DETECTION IN ENERGY OPTIMIZATION OF HIGH RACK STORAGE SYSTEMS

Cihan BAYRAKTAR*
Hadi GÖKÇEN**

DOI: 10.33461/uybisbbd.790369

Öz

Birimler arasında sağlıklı veri akışının sağlanması ile dijitalleşen üretim sistemleri ve bu dijitalleşme süreci doğrultusunda otomatikleşen zeki fabrika yapıları gün geçtikçe üretim endüstrisinde kendisine daha fazla yer bulmaktadır. Bu tür sistemler, üretim önemli gelişmeler ve teknolojik ilerlemeler sağlamış olsa da çeşitli sorunları da beraberinde getirmektedir. Bunlardan bir tanesi de otonom çalışan üretim sistemlerinde gerçekleşen bir anormal durumun hızlı bir şekilde tespit edilerek, çözüme kavuşturulması sürecidir. Bu kapsamda son zamanlarda anomali tespiti için çeşitli çalışmalar yapılmaktadır. Anomali tespiti konusunda en çok destek alınan alanlardan bir tanesi de makine öğrenmesi algoritmalarıdır. Bu çalışmada, yüksek depolama sistemlerinin enerji optimizasyonu hakkında uygulanmış bir prototip çalışmadan elde edilmiş olan iki farklı veri seti üzerinde çeşitli makine öğrenmesi algoritmalarının performansları test edilmiştir. Sonuç olarak, Yapay Sinir Ağları, C4.5 Karar Ağacı, Rastgele Orman ve k En Yakın Komşu algoritmaları ile oluşturulan öğrenme modelleri, test edilen veri setleri içerisindeki anomalileri tespit etme konusunda yüksek başarım oranı elde etmişlerdir. Özellikle bu algoritmalar içerisinde Rastgele Orman algoritması yaklaşık %98 seviyesindeki doğruluk performansı ile dikkat çekmiştir.

***Anahtar Kelimeler**: Anomali Tespiti, Sınıflandırma, Makine Öğrenmesi, Zeki Fabrikalar.*

Abstract

The production systems digitized by ensuring healthy data flow between the units and the smart factory structures that are automated in line with this digitization process find more and more places in the production industry. Although such systems have provided important developments and technological advances in production processes, they also bring with it various problems. One of these is the process of quickly detecting and resolving an abnormal situation occurring in autonomous production systems. In this context, various studies have been carried out recently for anomaly detection. One of the most studied areas for anomaly detection is machine learning algorithms. In this study, the performances of various machine learning algorithms were tested on two different data sets obtained from a prototype study on energy optimization of high storage systems. As a result, learning models created with Artificial Neural Networks, C4.5 Decision Tree, Random Forest and k Nearest Neighbor algorithms have achieved a high performance rate in detecting anomalies within the tested data sets. Among these algorithms, the Random Forest algorithm has attracted attention with its accuracy performance of approximately 98%.

***Keywords:** Anomaly Detection, Classification, Machine Learning, Smart Factories.*

* Öğr. Gör., Karabük Üniversitesi, Eskipazar Meslek Yüksekokulu, Bilişim Güvenliği Teknolojileri, Karabük, Türkiye, e-posta: cihanbayraktar@karabuk.edu.tr, ORCID: 0000-0003-4321-5485
** Prof. Dr., Gazi Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği, Ankara, Türkiye, e-posta: hgokcen@gazi.edu.tr, ORCID: 0000-0002-5163-0008

Geliş Tarihi / Received: 04.09.2020
Kabul Tarihi / Accepted: 16.11.2020

1. GİRİŞ

Endüstri 4.0, 2011 yılında üniversiteler ve özel şirketler ile birlikte Alman Federal Hükümeti tarafından bir girişim olarak icat edilmiştir. Stratejik bir program olarak ortaya çıkan bu programın amacı, endüstrinin üretkenliği, etkinliği ve verimliliğini arttırmak ve gelişmiş üretim sistemleri ortaya çıkartmaktır. Bu yapı, ürün yaşam döngüsüne katkı sağladığı bilinen bir dizi teknolojinin bir çatı altında bir araya getirilerek, ortak bir yapı içerisine entegre edilmesini kapsamaktadır. Endüstri 4.0, gelişmiş üretim veya zeki üretim yapısında, esnek hatların oluşmasına imkan tanıyan ve bu sayede, çok çeşitli ürün türü ve değişen şartlar doğrultusunda üretim süreçlerinin otomatik ayarlandığı bir sistem olarak kullanılmaktadır (Frank vd., 2019).

Endüstri 4.0'ın diğer endüstri devrimleri içerisinde planlı olarak gerçekleşen ilk devrim olacağı belirtilmektedir. Yeni nesil endüstri, dijitalleşen üretim sistemlerinin çıktıları ve bileşenleri tarafından şekillendirilecektir. Bu durum sayesinde, üretim sistemlerinde kullanılan tüm fiziksel bileşenler, makineler arası iletişim sistemine entegre edilecektir. Özellikle üretim sistemlerinde uygulanacak olan dijitalleşme süreci, optimum çalışma, kişiselleştirilmiş ürünler ve esnek üretim yapılarının ortaya çıkmasını sağlayacaktır (Riordan vd., 2019).

Büyük veri, nesnelerin interneti (IoT), siber fiziksel sistemler gibi yenilikçi kavramların yükselişleri ve çeşitli teknolojilerin geliştirilmesi yeni dönemde endüstri yapısını üst seviyelere taşımıştır. Bunların sonucunda zeki üretim sistemlerinin oluşturulmasını hedefleyen endüstriyel nesnelerin interneti (IIoT) yapısı ile zeki fabrikalar ortaya çıkmıştır. IIoT sayesinde zeki fabrikalar, bilgilerin bağımsız olmadığı grup etkileşimlerini gerçekleştirmektedirler. Birimler arasında bilgilerin kaynaşması ve çarpışması aracılığı ile zeki üretim süreçlerinin geliştirilmesi sağlanabilmektedir (Wan vd., 2019). Zeki fabrikalar, ürünlerin kısa yaşam döngülerinden, deneyimli çalışan eksikliğinden, ülkelerce yürütülen çeşitli çevre düzenlemelerinden ve sürekli olarak artan müşteri taleplerinden kaynaklanan sorunların, hızlı ve hatasız bir şekilde çözüme kavuşturulabilmesi için geliştirilmektedir (Yoon vd., 2019).

IIoT, zeki fabrika sistemlerinde temel ekipmanların yapıya entegre edilmesinde kullanılan bir teknolojidir. Bu şekilde üretim sistemi, algılama, ara bağlantı sağlama ve verilerin entegrasyonunu gerçekleştirme yeteneğine sahip olur. Verilerin analiz edilmesi ve bilimsel karar verme süreçleri, zeki fabrikalar bünyesinde, üretim planlanmasını, ekipmanların verimli kullanımını ve kalite kontrol süreçlerinde kullanılmaktadır. Ayrıca sistem verilerinin yerel bir veri tabanından, bulut sistemine yüklenmesi için internet de ciddi anlamda kullanılmaktadır. Zeki fabrikalar, insan ve makinenin etkileşimi sayesinde, küresel iş birliğine dayalı zeki üretim sistemlerini inşa etmektedirler (Chen vd., 2017).

Zeki fabrikalar, gün geçtikçe daha da karmaşık hale gelen bir dünyada, dinamik ve hızlı değişen şartlara sahip bir üretim sistemi için, ortaya çıkabilecek sorunları çözebilecek esnek bir üretim sistemi çözümüdür. Bu çözüm aynı zamanda, gereksiz iş gücü ve kaynak israfının önüne geçebilmek amacıyla kullanılması gereken yazılım ve donanım birimlerinin kombinasyonunu da ilgilendirmektedir. Ayrıca zeki sistemlerin gerçekleştirmesi gereken görevlerden biri olan endüstriyel ortam ve çevre arasındaki bağlantının da sağlıklı bir şekilde kurulması ve yönetilmesi konuları ile de ilgilenmektedir (Radziwon vd., 2014).

Zeki fabrika sistemleri için önemli olan sorunlardan bir tanesi anomali tespiti aşamasıdır. Zeki fabrikaların sahip olduğu karmaşık yapı, sistemlerin istenmeyen durumların oluşmasına sebep olabilmektedir. Son zamanlarda, özellikle güvenlik açıkları ve anomali konusu, zeki fabrikalar için fenomen halini almıştır (Hasan vd., 2019). Anomali tespiti için çalışma yapılan sistemlerde, bakım sıklığını en düşük seviyeye çekebilmek için tahmini bakım çalışmaları yapılabilir ve üretim kaynaklarının verimli kullanılması kapsamında uygulanan çalışmalarda önemli ölçüde maliyet avantajı sağlanabilir. Ayrıca fabrika içi üretim kapasitesi ve sistemin karmaşıklık seviyesi arttıkça, karşılaşabilecek olan sorunların çeşidi ve miktarı da artış gösterecektir. Oluşması muhtemel bu

sorunların çözümlerinde, sorunu erken tespit etmek ve maliyetleri en aza indirebilmek için, üretim sistemi içerisinde çalışan cihazların anormal davranışlarının analizlerinin yapılması ve tespit edilmesi gerekmektedir. Böylece üretim sisteminde yaşanacak süreç gecikmeleri ve zararların daha hızlı önüne geçmek mümkün olabilecektir (Hsieh vd., 2019).

Endüstri 4.0 devriminde, zeki fabrikalar tarafından temsil edilen fiziksel sistemler ve bilişim teknolojilerinin etkileşimi, birbirine bağlı olan tüm sistemler arasında çok büyük miktarlarda gerçek zamanlı veri alışverişine imkan sağlamıştır. Bu sayede, üretim alanında anomali tespit sistemleri için gereklilik ortaya çıkmıştır. Anomali tespit sistemlerinde, üretim sistemi içerisinde çalışan cihazlardan toplanan veriler, bilişim sistemleri içerisine dahil edilmektedir. Sisteme aktarılan veriler üzerinden anlık değerlendirme yapılarak, herhangi bir anomali tespit edildiği takdirde, sistemde oluşacak düşük performansı ve yüksek hata ihtimallerini hızlı bir şekilde engellemek için operatörlere gerekli bilgilerin aktarım işlemleri sağlanmaktadır. Bu sayede yüksek oranda kesintisiz ve performanslı bir üretim sisteminin işleyişi mümkün olabilmektedir (Bagozi vd., 2017).

Bu çalışmada, zeki fabrikaların üretim sistemlerinde, anomali tespiti konusunda makine öğrenmesi algoritmalarının nasıl bir başarım gerçekleştirdiklerinin ölçülmesi ve uygun algoritmanın tespit edilmesi amaçlanmıştır. Bu kapsamda da kaggle sisteminden temin edilmiş olan açık kaynak kodlu ve yüksek depolama sistemlerinde enerji optimizasyonun sağlanması üzerine yapılmış bir çalışmadan elde edilmiş veri setleri kullanılmıştır. Çalışmanın ikinci bölümünde, anomali tespiti üzerine yapılmış olan çalışmalar hakkında bilgi verilmiştir. Üçüncü ve dördüncü bölümlerde ise, veri setleri üzerinde makine öğrenme algoritmalarının anomali tespiti konusundaki başarımları ölçülmüş ve birbirleri arasında kıyaslanarak, sonuçlar yorumlanmıştır.

2. İLGİLİ ÇALIŞMALAR

2016 yılında gerçekleştirilen çalışmada, hibrit üretim sistemlerinde kullanılabilecek olan otomatik öğrenme özelliğine sahip anomali tespit algoritması önerilmiştir. Sistem içerisinde uygulanan gözlemlerden tespit modelini oluşturmak için derin öğrenme teknikleri ve zamanlı otomata sistemlerinin birleşiminden yararlanılmıştır. İki adet gerçek sistem dahil olmak üzere çeşitli veri setleri üzerinde test ettikleri algoritmanın umut verici sonuçlar verdiği açıklanmıştır (Hranisavljevic vd., 2016). 2017 yılında yayınlanmış olan devam çalışmasında da hibrit üretim sistemlerinde anomali tespiti için, denetimsiz ve parametrik olmayan bir yaklaşım oluşturulmuştur. Bu yaklaşım ile normal şartlarda anomali tespit uygulaması mümkün olmayan üretim sistemlerinde, hibrit zamanlı otomata kullanımına izin vererek anomali tespiti yapılmasını sağlayan, kendi kendini düzenleyen haritalar ve havza dönüşümleri kullanılmıştır (Birgelen ve Niggeman, 2017).

2018 yılında yapılmış olan bir çalışmada, IoT sisteminde hizmet içi servislerin iletişim ve çalışma yapısını öğrenen ve kendini sürekli güncel tutan, kaynak verimli bir yaklaşım önerilmiştir. Önerilen bu yaklaşımın, düğümler arasındaki süreç iletişiminde akan verileri analiz ederek, öğrenmiş olduğu model doğrultusunda anomali tespiti yapabildiği belirtilmiştir. Çalışma sayesinde IoT sistemlerinin güvenlik seviyelerinin daha üst noktalara ulaşabildiği sonucuna varılmıştır (Pahl ve Aubet, 2018).

2019 yılında yayınlanmış olan bir çalışmada, nesnelerin interneti üzerine gerçekleşebilecek siber saldırılar dolayısı ile oluşabilecek anomalilerin tespit edilmesi noktasında çeşitli makine öğrenmesi algoritmalarının performans karşılaştırmaları yapılmıştır. Yapılan ölçümler sonucunda, Karar ağacı, rastgele orman ve yapay sinir ağları algoritmalarının %94 gibi yüksek başarıya ulaştığı, performans bakımından ise rastgele orman algoritmasının öne çıktığı tespit edilmiştir (Hasan vd., 2019).

Bir başka çalışmada, üretim sistemi üzerinden bulunan cihazlardan toplanan gerçek veriler kullanılarak, zeki üretim sistemlerinde anomali tespiti için kullanılabilecek bir algoritma

önerilmiştir. Üretim hattından elde edilen çok değişkenli sensor veri setlerinde bulunan sınırlı ve düzensiz anomali verilerinin ortaya çıkartılabilmesi için, otomatik kodlayıcıya dayalı denetimsiz gerçek zamanlı anomali algılama algoritması kullanılmıştır. Sonuç olarak önerilen bu algoritmanın anomali tespitinde %90 başarım oranı elde ettiğini belirtilmiştir (Hsieh vd., 2019).

Wang ve diğerleri tarafından anomali tespiti için derin öğrenme yapıları üzerine yapılan çalışmada araştırmacılar, derin öğrenme tabanlı oluşturulan anomali tespit sistemlerinin daha iyi anlaşılmasını amaçlamışlardır. Çalışmada ilk etapta derin öğrenme yapılarından önce uygulanan anomali tespit teknikleri açıklanmış ve sonrasında günümüzde uygulanan yüksek teknolojiye sahip derin öğrenme tabanlı anomali tespit tekniklerinin, öncesinde kullanılan geleneksel algoritmaların sorunlarını aşma konusunda kullandıkları teknikleri tartışmışlardır (Wang vd., 2020).

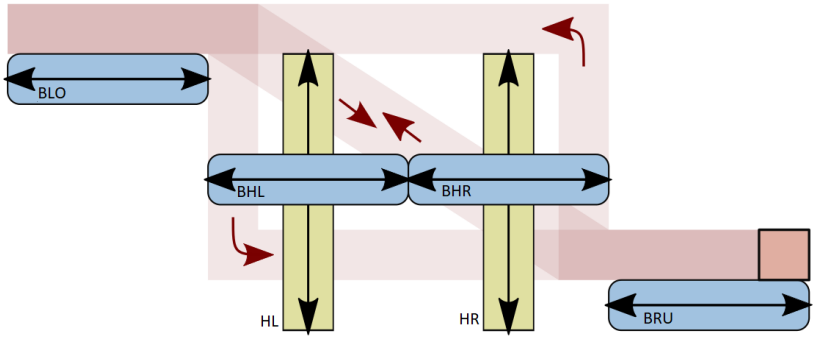
3. DENEYSEL ÇALIŞMA

Sistemin tamamı çeşitli bağımsız süreçlerden oluşmaktadır. Buradaki ilk aşama verilerin bir araya getirilmesidir. Veriler, dikkatli bir şekilde toplanıp incelenerek, uygun veri tipleri elde edilmeye çalışılmaktadır. Bir sonraki süreçte ise veri üzerinde önişleme adımları uygulanmaktadır. Önişleme adımları, verinin içerisinde bulunan gürültünün temizlenmesi, eksik verilerin tamamlanması, veri üzerinde dönüştürme ve birleştirme işlemlerinin uygulanmasından oluşmaktadır. Önişleme uygulamalarının ardından veri, artık sınıflandırma algoritmalarının testi için kullanılabilir duruma gelmiş olacaktır. Bu kısımda veri setlerinin analizinde 10 Fold Cross Calidation yöntemi kullanılmıştır. Test edilen her algoritma, öğrenme kümesini kullanarak kendi öğrenme modelini oluşturacak, sonrasında ise bu modeli test kümesi üzerinde sınayarak başarım derecesini ölçecektir. Bu çalışma kapsamında farklı sınıflandırma algoritmaları kullanılmıştır. Bunlar; Lojistik Regresyon, Naive Bayes, Destek Vektör Makineleri, Karar Ağaçları, Rastgele Orman, k En Yakın Komşu ve Yapay Sinir Ağları algoritmalarıdır.

3.1. Veri Setinin Oluşturulması ve Tanımlanması

Çalışmada kullanılan açık kaynak veri seti, Hranisavljevic ve diğerleri tarafından oluşturulmuş ve bu çalışma için kaggle ortamından çekilmiştir (Hranisavljevic vd., 2016; Hranisavljevic vd., 2018). İlgili veri setinin oluşturulabilmesi amacıyla, dört adet kısa konveyör bandından (BLO, BHR, BHL ve BRU) ve iki raydan (HL ve HR) oluşan bir yüksek raflı depolama sistemi oluşturulmuştur. Oluşturulan yüksek raflı depolama sisteminin görsel modeli, Şekil 1'de verilmiştir. Ortada bulunan BHL ve BHR konveyör bantları, raylar üzerinde dikey yönlü hareket gerçekleştirmektedirler. Diğer bantlar ise sabittir. Bu sistem, iki nokta arasında paket taşımak için oluşturulmuştur. Sistemin oluşturulan ilk versiyonunda orta konveyör bantları dikey hareket halinde iken yatay yönlü paket taşımayacak şekilde ayarlanmış ve bu modda çalıştırılarak ilk veri seti elde edilmiştir. İkinci versiyonda ise sistem optimize edilmiş ve orta bantlar dikey hareket ile aynı anda yatay yönlü paket taşıma işlemini de yapacak şekilde güncellenmiş ve ikinci veri seti oluşturulmuştur (Hranisavljevic vd., 2018). Bu çalışma bünyesinde, elde edilmiş olan veri setlerinin ikisi ile ayrı ayrı sınıflama algoritmaları uygulanmış, optimizasyon işlemi uygulanmış veri seti ile optimize edilmemiş veri seti arasındaki fark da gözlenmiştir.

***Şekil 1**: Yüksek Raflı Depolama Sisteminin Görsel Modeli* (Hranisavljevic vd., 2018).



Tablo 1'de verilmiş olan düzene göre, veri setlerinin her birinde 19 adet nitelik ve bir adette sınıf niteliği bulunmaktadır. Bu niteliklerden ilki işlem süresinin gösterildiği zaman damgası niteliğidir. Diğerleri ise yapı içerisindeki her bir elemanın (Dört adet Konveyör ve iki adet ray) mesafe, güç ve voltaj sinyallerinden oluşmaktadırlar (Birgelen ve Niggeman, 2017).

Tablo 1: Veri Setleri Nitelik Tanımları.

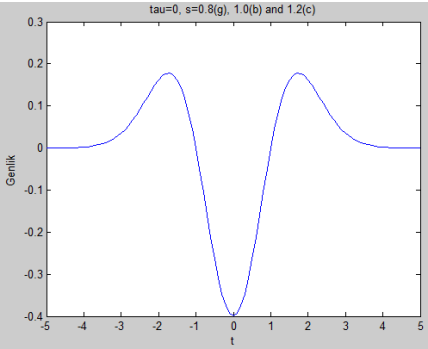
Nitelik Adı	Açıklaması	Standart Veri Seti Değer Aralığı	Optimize Edilmiş Veri Seti Değer Aralığı
TimeStamp	Saniye Cinsinden Süre	0 – 18,851	0 – 14,246
I_w_BLO_Weg	Sol Üst Konveyör Mesafe Bilgisi	(-315,836) – 739,2	(-3,4,264) – 1011,098
O_w_BLO_Power	Sol Üst Konveyör Güç Bilgisi	0 – 34817,661	(-82,014) – 30140
O_w_BLO_Voltage	Sol Üst Konveyör Voltaj Bilgisi	(-113,02) – 179,025	0 – 60
I_w_BHL_Weg	Orta Sol Konveyör Mesafe Bilgisi	(-548) – 1301,892	(-895,2) – 1130,4
O_w_BHL_Power	Orta Sol Konveyör Güç Bilgisi	(-4314,545) – 48719,681	(-8990,991) – 39838,352
O_w_BHL_Voltage	Orta Sol Konveyör Voltaj Bilgisi	(-22,111) – 66	(-43,87) – 105
I_w_BHR_Weg	Orta Sağ Konveyör Mesafe Bilgisi	(-1322) – 621,4	(-1322) – 1015,3
O_w_BHR_Power	Orta Sağ Konveyör Güç Bilgisi	0 – 32536	(-241,091) – 41507,7
O_w_BHR_Voltage	Orta Sağ Konveyör Voltaj Bilgisi	0 – 67,2	(-0,955) – 119,6
I_w_BRU_Weg	Sağ Alt Konveyör Mesafe Bilgisi	(-661,952) – 688,8	(-855) – 755,851
O_w_BRU_Power	Sağ Alt Konveyör Güç Bilgisi	0 – 62674	0 – 35008,471
O_w_BRU_Voltage	Sağ Alt Konveyör Voltaj Bilgisi	0 – 75,4	0 – 72,8
I_w_HL_Weg	Sol Ray Mesafe Bilgisi	(-1082,9) – 101,118	(-1151,204) – 186,85
O_w_HL_Power	Sol Ray Güç Bilgisi	0 – 41789,8	0 – 41940,6
O_w_HL_Voltage	Sol Ray Voltaj Bilgisi	0 – 596,4	0 - 279
I_w_HR_Weg	Sağ Ray Mesafe Bilgisi	(-1032,2) – 0	(-833) – 0
O_w_HR_Power	Sağ Ray Güç Bilgisi	0 – 42895,835	0 – 39060,793
O_w_HR_Voltage	Sağ Ray Voltaj Bilgisi	0 – 543	0 – 280
Sınıf	Anomali Bilgisi	0 veya 1	0 veya 1

Yüksek raflı depolama sisteminin çalıştırılması ile elde edilen veriler, anomali tespit işlemlerinde kullanılabilmesi amacıyla iki boyuta indirgenmiştir. Eğitim amaçlı yapılan ilk gözlemler, her değerlendirme gözleminde mesafeyi hesaplamak için referans değerleri sağlamaktadır. Mesafe değeri belirli bir eşik değerini aşan veriler anomali olarak işaretlenmiştir. Anomali durumunu gösteren eşik değerinin hesaplanmasında ise Meksika Şapkası Dalgacık yöntemi kullanılmıştır. Meksika Şapkası Dalgacığı, Gauss Fonksiyonunun normalizasyon

işleminden sonra elde edilen versiyonunun ikinci türevinin alınış halidir. Şekil 2'de görüldüğü üzere, eğri biçimi Meksikalıların giydiği şapkaya benzediğinden dolayı bu isimle anılmaktadır. Matematiksel ifadesi, aşağıdaki eşitlikte gösterilmektedir (Şeker vd., 2018).

$$\varphi(t,\tau,s) = \frac{[\left(\frac{(t-\tau)}{s}\right)^2 - 1]exp\{(\frac{(t-\tau)}{s})^2 * (-0{,}5)\}}{\sqrt{2\pi} * s^3} \quad (1)$$

Şekil 2. Meksika Şapkası Dalgacığı Grafiği



Eşik değeri için en uygun seviyesinin belirlenmesi önemlidir. Çünkü, eğer eşik değeri çok yüksek olursa, (Örn: %100) normal değerlerin anomali olarak algılanma olasılığı artacaktır. Eşik değeri çok düşük olursa, (Örn: %25) o zaman da algoritmalar anomaliyi tespit edemeyeceklerdir. Bu çalışmada kullanılan veri setlerinde işaretlenmiş olan anomaliler, %60 eşik değeri ile belirlenmişlerdir (Birgelen ve Niggeman, 2017).

Çalışma kapsamında kullanılan veri setlerinden birincisi, ortadaki konveyör bantlarının dikey hareket halinde iken yatay yönlü paket taşıma işlemi yapmadıkları standart çalışma süreçleri ile elde edilen verilerden oluşmaktadır. Bu veri setinde, toplam 23645 satır veri bulunmaktadır. Bu verilerden 5670 adedi anomali olarak işaretlenmiş, kalan 17975 adet veri ise normal süreç verileri olarak kaydedilmiştir.

İkinci veri seti ise ortada bulunan konveyör bantlarının dikey hareket halinde iken, yatay yönlü paket taşıma sürecini de gerçekleştirdikleri optimize edilmiş süreçlerden elde edilen verilerden oluşmaktadırlar. Bu veri setinde, toplam 19634 satır veri bulunmakta, bunlardan 4517 adedi anomali olarak işaretlenmiş, diğer 15117 adedi ise normal süreç verileri olarak kaydedilmiştir.

3.2. Veri Ön İşleme

Çeşitli çalışmalar kapsamında kullanılmak üzere elde edilen veri setlerinde, bazı verilerde eksiklikler, hatalar, tekrarlar veya anlamsızlıklar bulunabilmektedir. Bundan dolayı, veri setleri üzerinde çalışma yapmaya başlamadan önce, çeşitli veri düzenleme süreçlerinden geçirmek önem arz etmektedir. Bu süreçler, kayıp verilerin düzenlenmesi, gürültünün ortadan kaldırılması, bütünleştirme, dönüştürme ve azaltma şeklinde isimlendirilebilmektedir (Aydemir, 2019). Dolayısı ile araştırmacının elindeki veri setinin ihtiyaçları doğrultusunda, anılan düzenleme işlemlerinden uygun olanları değerlendirmesi gerekmektedir.

Kaggle ortamından temin edilen iki adet veri setleri üzerinde yapılan gözlemlerde, herhangi bir eksik ve anlamsız veriye rastlanmamıştır. Veri setlerinin her birinde bir adet sınıf etiketi, 19 adet ise nitelik etiketi bulunmaktadır. Nitelik etiketlerinin tamamı, nümerik verilerden oluşmaktadır. Sınıf etiketi ise verilerde anomali var olup olmadığını gösterdiği için, 1 veya 0 değerlerini içerecek şekilde nominal olarak işaretlenmiştir.

3.3. Teorik Kavramlar

Çalışma kapsamında birçok sınıflandırma algoritması kullanılmış ve karşılaştırma işlemleri yapılmıştır. Kullanılan algoritmalar ile ilgili açıklamalar alt başlıklarda ifade edilmiştir.

3.3.1. Naive Bayes (NB)

İstatistiksel bir sınıflandırma algoritması olan Naive Bayes (NB), arka planda istatistiksel değerlere göre farklılık gösterebilen bir çalışma sistemine sahiptir. Bu sebeplerden dolayı dinamik olarak çalışan sistemler üzerinde kullanımı esnasında, tekrar tekrar hesaplama işlemlerinin gerçekleştirilmesini gerektiren bir algoritmadır (Şeker, 2016).

NB sınıflandırma tekniği, Bayes kuralı ile birlikte karar ağaçları modeli birleştirilerek elde edilmiş bir tekniktir. NB Algoritması, örneği verilmiş olan her sınıfın olasılık değerini hesaplamak amacıyla Bayes kuralını kullanmaktadır (Akçetin ve Çelik, 2014). Makine öğrenmesi uygulamalarında sıkça karşılaşılan bir sınıflandırma tekniği olan NB, koşullu olasılık hesaplamaları üzerine çalışan ve Bayes kuralının en basit hali olarak nitelendirilen bir algoritmadır (İşçimen vd., 2014). NB, sınıflandırma teknikleri içerisinde en kısıtlayıcı alanda yer almaktadır. Bu algoritmada örnek verilerin hangi sınıfa ait oldukları bilinmektedir. Genel anlamda metin sınıflandırılmasında üstün başarı gösterdiği tespit edilmiştir (Nizam ve Akın, 2014). NB algoritması, koşullu sınıf bağımsızlığının varsayımı üzerinde durmaktadır. Yani, herhangi bir değer niteliğinin, diğer niteliklerin değerlerinden bağımsız olduğu düşünülür. Pratikte nitelikler arasında bir miktar bağımlılık olsa dahi, hesaplamaların kolaylaştırılması amacıyla teoride bu varsayım uygulanmaktadır. Eğer uygulanan varsayımlar doğru olursa, NB algoritması diğerlerine göre en iyi sonucu veren algoritma olacaktır. Ayrıca işleme tabi tutulan nitelikler arasında öneme göre bir derecelendirme yapılmaz ve sınıfın tahmin edilmesinde tüm niteliklerin aynı derecede önemli olduğu kabul edilir (Han ve Kamber, 2006).

Genel anlamda NB tekniği, X dizisi içerisindeki her bir verinin, C sınıfına ait olup olmama olasılığını hesaplamak için tercih edilmektedir (Hand vd., 2001).

$$P(h1|xi) = \frac{P(xi|h1)P(h1)}{P(xi|h1)P(h1)+P(xi|h2)P(h2)} \quad (2)$$

Eşitlik (2)'de P(h1) ifadesi, h1 hipotezi ile birlikte ön olasılık olduğu zaman, p(h1|xi) ifadesi sonraki olasılık olarak değerlendirilmektedir (Patil ve Sherekar, 2013).

Ayrıca eşitlik (3) ile P(h1|xi) değeri maksimize edilmektedir. Maksimize edilmiş olan P(h1|xi) değerinin C sınıfı, Bayes teoremine göre maksimum sonraki olasılık olarak ifade edilmektedir (Han ve Kamber, 2006).

$$P(h1|xi) = \frac{P(xi|h1)P(h1)}{P(xi)} \quad (3)$$

3.3.2. Yapay Sinir Ağları (YSA)

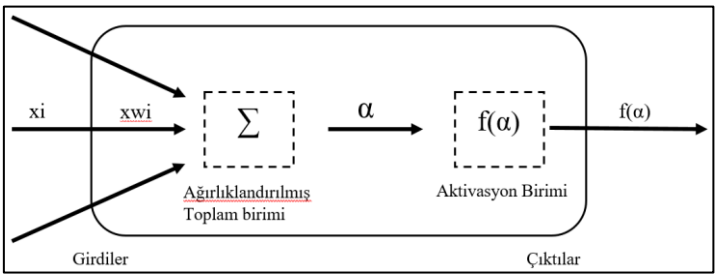
Yapay Sinir Ağları (YSA), ağırlıklı bağlantıları kullanarak birbiri ile bağlantı kurmuş olan elemanlardan oluşmaktadır. Ayrıca bu elemanların, her biri kendilerine ait, paralel ve dağıtılmış bilgi işleme yeteneğine sahip bellekleri bulunmaktadır. Farklı bir ifade ile YSA, biyolojik sinir sisteminin kopyası gibi çalışması amaçlanmış bir bilgisayar programıdır. YSA, bu özelliği sayesinde kendi kendine öğrenen bir yapıya sahiptir. Öğrenmenin yanında ezberleme, bilgiler arasındaki ilişkiyi ortaya çıkartma gibi yetenekleri de bulunmaktadır. Ayrıca tüm bunları yapabilmesi konusunda yazılımcının geleneksel yeteneklerine muhtaç değildir (Elmas, 2016).

YSA, yapay zeka çalışmalarının gelişmesine katkı sağlayan alanlardan bir tanesidir. Buna dayanarak YSA'nın, öğrenme yeteneğine sahip sistemlerin başında gelen yapay zeka teknolojilerinin bir parçası olduğu düşünülebilir. YSA için, insan beyninin temel elemanlarından olan nöronların çalışma prensiplerini kopyalamaya çalışarak, gerçek sinir sistemi yapısının bir simülasyonunu oluşturmaya yarayan programlar olduğu söylenebilir (Aydemir, 2019).

YSA, insan beyninde bulunan nöronlar gibi çalışan yapay nöronlar aracılığı ile örnekler üzerinde yeterli incelemeleri yaparak değerli olan bilgiyi ortaya çıkartmak için kullanılmaktadır. Bu yapay nöronlar, problemlerin çözüm aşamalarında kendi öğrendiklerini kullanarak karar verebilme yeteneğine sahiptirler. Kısaca YSA, çeşitli geometrik şekillere sahip yapay nöronların arasında kurulan bağlantı ile oluşan ağ yapıları olarak ifade edilebilmektedir. Bahsi geçen ağ yapıları oluşturulduktan sonra, yeni gelen verilerin sınıflandırılması için kullanılmaktadırlar (Staub vd., 2015). YSA sıklıkla, doğrusal olsun veya olmasın herhangi bir problem hakkında girdi olarak kullanılan veriler ile çıktı olarak elde edilmesi gereken veriler arasında gerekli bağlantıyı kurarak sonraki uygulamalar hakkında sonuçlar elde edebilmek amacıyla kullanılmaktadır (Yakut vd., 2014).

Şekil 3'te gösterilen, basit problemlerin çözümünde kullanılabilecek olan sinir ağı, girdi ve çıktı nöronlarından oluşan tek katmanlı ağlar olarak ifade edilmektedir.

Şekil 3: Tek Katmanlı Yapay Sinir Ağı Yapısı (Yakut vd., 2014)

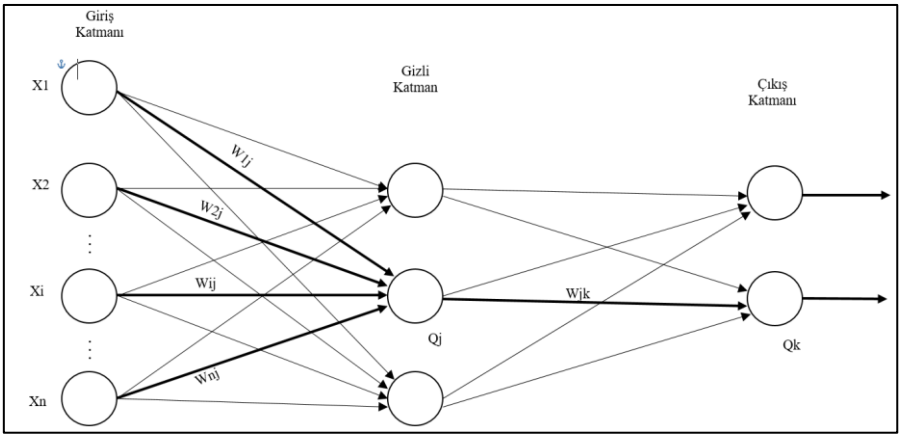


Bu tip ağlarda tüm girdi ve çıktı nöronları bir veya birden fazla olabilir ve tüm girdi nöronları, tüm çıktı nöronlarına ağırlıklandırılarak bağlanmaktadır. Ağırlıklandırılmış bağlantı, eşik değeri ile kontrol edilerek aktivasyon fonksiyonun çalıştırılması ve çıkış değerinin hesaplanması hedeflenmektedir. Bu olay matematiksel olarak şu şekilde ifade edilir (Yakut vd., 2014).

$$f(\alpha) = \sum_{i=1}^{m} wixi + \theta \quad (4)$$

Karmaşık problem çözümlerinde ise tek katmanlı ağlar yeterli olamayacağından dolayı girdi ve çıktı katmanlarının arasında gizli katmanların devreye girdiği, Şekil 4'te görünen çok katmanlı ağlar kullanılmaktadır.

Şekil 4: Çok Katmanlı Yapay Sinir Ağı Yapısı (Han ve Kamber, 2006)



Bu yapıda ağ, bir veri grubunun sınıf etiketini tahmin edebilmek için katmanlar arasında görev yapan ağırlıkların gerçek değerlerini öğrenme yoluna gider. Giriş katmanında bulunan nöronlardan gelen giriş verileri ağırlıklandırılarak, gizli katmanda bulunan nöronların değerleri hesaplanır. Sonrasında ise gizli katmanda bulunan değerler yine ağırlıklandırılarak, çıkış katmanı verisinin hesaplanmasına çalışılır (Han ve Kamber, 2006). Öğrenme modelinin oluşturulması aşamasında, elde edilen çıktı katmanı verisi ile beklenen çıktı katmanı verisi karşılaştırılarak hata miktarı hesaplanır. Ortaya çıkan hata miktarı doğrultusunda geri besleme yapılarak hata katsayıları hesaplanmaktadır ve sonrasında elde edilen katsayılar ile nöron bağlantılarında kullanılan ağırlık miktarları değiştirilmektedir. Sistem, hesaplanan çıktı katmanı verisi ile beklenen çıktı katmanı verisi arasındaki hata miktarı en aza indirilene kadar bu işlemleri tekrarlamaya devam eder (Arı ve Berberler, 2017).

3.3.3. Lojistik Regresyon (LR)

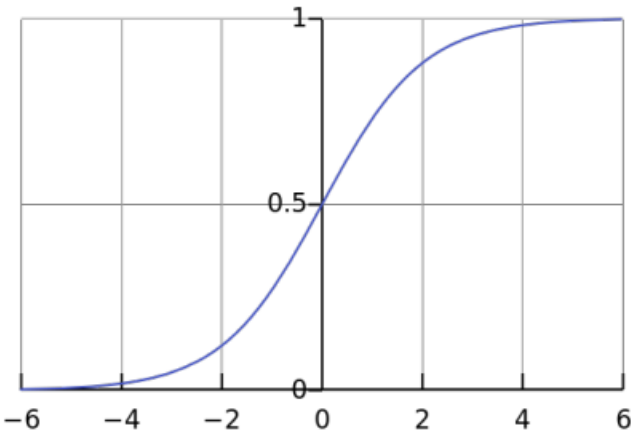
Lojistik Regresyon (LR) veri seti içerisinde bulunan tüm değişkenleri sayısal kabul eden ve normal bir dağılıma sahip olan, ikili sınıflandırma algoritması olarak tanımlanmaktadır. Bu varsayımda belirtilen noktaya rağmen, LR ile normal dağılım göstermeyen veriler üzerinde dahi iyi sonuçlar alınabilmektedir. LR algoritması, doğrusal olarak bir regresyon fonksiyonun içerisinde birleştirilmiş ve bir lojistik fonksiyon kullanılarak dönüştürülmüş her giriş değeri için bir katsayı öğrenmeye dayalı bir sistem üzerine kurulmuştur. Hızlı ve basit bir sisteme sahip olmasına rağmen, bazı problemler üzerinde son derece etkili sonuçlar vermektedir. LR sadece ikili sınıflandırma modellerini desteklemektedir (Brownlee, 2019).

LR, düzeltilmiş olasılık oranları hakkında çıkarım sağlamak için geliştirilmiş standart bir öğrenme algoritmasıdır (Mansournia vd., 2018). LR, Veri setinin kalitesine bağımlı olarak ayrıcı özelliğine sahip bir modeldir. Modelin belirlenmesinde, özellik değerleri (X1, X2,…,Xn), ağırlık değerleri (W1, W2,…,Wn), sapma değerleri (b1, b2,…,bn) ve sınıflar (1 / 0) dikkate alındığında eşitlik (5) kullanılabilmektedir (Hasan vd., 2019):

$$Tahmin\ Edilen\ Değer = p(y = C|X; W, b) = \frac{1}{1+\exp(-w^{transpose}\ X-b)} \quad (5)$$

LR, logaritmik ilerleme gerçekleştiren, logit bir fonksiyondur. LR, yapı itibari ile Şekil 5'teki grafikte gösterilen eğriye en iyi şekilde benzeyecek olan değerleri tespit etmeye çalışmaktadır (Şeker, 2016).

Şekil 5. Lojistik Regresyon Grafiği



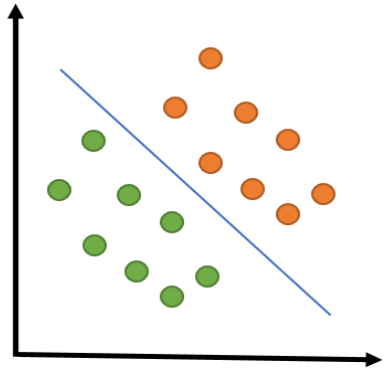
3.3.4 Destek Vektör Makineleri (DVM)

Destek Vektör Makineleri algoritması, LR algoritması gibi ayrımcı bir model olarak çalışmaktadır. Regresyon, aykırı veri tespiti ve sınıflandırma işlemleri için denetimli bir öğrenme sistemi sunmaktadır (Hasan vd., 2019). DVM, 1995 yılında Vladimir Vapnik tarafından tanıtılan, sınıflandırma ve örüntü tanıma süreçleri için kullanılabilen, basit ve verimli bir algoritmadır. DVM çalışma sisteminin ana amacı, hiper düzlemlerin ve sınırların ortaya çıkartacak fonksiyonların elde edilmesidir. Oluşturulan hiper düzlemler, istatistiksel öğrenmeyi kullanmak amacı ile belirli algoritmalar ile eğitilerek, giriş verisi noktalarının farklı kategorilere ayrıştırılması sağlamak için kullanılmaktadır (Jain vd., 2018).

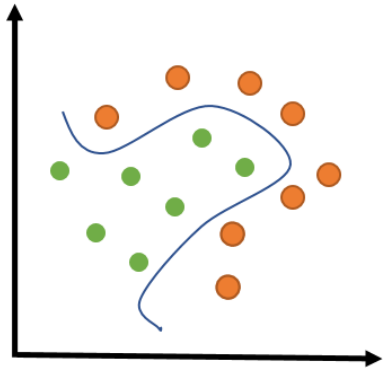
Destek Vektör Makineleri, prensip olarak istatistiksel öğrenme yöntemleri ve yapısal risklerin minimize edilmesine dayalı bir öğrenme algoritmasıdır. Hızlı öğrenme kapasitesine sahip büyük giriş verilerini kullanabilmesi, algoritmanın temel avantajını ortaya çıkartmaktadır. Bu sayede DVM algoritması ile doğrusal olmayan yüksek boyutlu veri modelleme sorunlarına çözüm getirilebilmektedir (Feizizadeh vd., 2017).

DVM, çoğunlukla sınıflandırma işlemlerinde kullanılan bir algoritmadır. DVM algoritmasının çalışma sisteminde, n bağımsız değişken sayısı olmak üzere tüm veri noktalarının n-boyutlu uzaydaki koordinatları değişken değerleri olarak belirlenmektedir. Bir üst aşamaya geçildiğinde ise iki sınıfı birbirinden ayıran iki boyutlu hiper düzlem tespit edilir ve sınıflandırma işlemi, Şekil 6'da gösterildiği gibi gerçekleştirilir. Fakat, elde edilen tüm verilerin doğrusal hiper düzlemler ile sınıflandırılması mümkün değildir. Doğrusal olmayan veri türleri, DVM sınıflandırılmasında kernel fonksiyonu kullanılarak yüksek boyutlu uzayda oluşturulan doğrusal olmayan bir hiper düzlem ile Şekil 7'de görselleştirildiği gibi birbirlerinden ayrılabilir duruma getirilebilmektedirler. Bu konumda uygulanan kernel fonksiyonu, doğrusal modda çalışan bir sınıflandırma algoritmasının, doğrusal olmayan bir problemi çözmesini sağlayacak olan kernel hilesi anlamında kullanılmaktadır (Gürsakal, 2018).

Şekil 6: Doğrusal Hiper Düzlem ile Sınıflandırma



***Şekil 7**: Doğrusal Olmayan Hiper Düzlem ile Sınıflandırma*



3.3.5. Karar Ağacı (KA)

Karar Ağaçları (KA) sınıflandırma ve regresyon problemlerinin çözümlerini destekleyebilmektedir. Veri örneklerini değerlendirmek amacıyla bir ağaç yapısının oluşturulması sistemine dayanmaktadır. Ters çevrilmiş bir ağacın kökü ile yapı başlar ve bir tahmin sonucuna ulaşılana kadar aşağıya doğru devam eder. KA, oluşturulması aşamasında, doğru tahminlere ulaşılabilmesi amacıyla en iyi ayırıcılık özelliğine sahip olan niteliğin tespit edilmesi süreçleri gerçekleştirilmektedir (Brownlee, 2019).

Dağılım tabanlı tahmin etme işlemlerinde, giriş verilerinin tamamı üzerinde bir modelin geçerli olduğu varsayılır ve veri setine ait parametrelerin öğrenilebilmesi için de bütün veri seti kullanılır. Öğrenme işleminin sonrasında, test verilerinde sistemin sınanması aşamasında da aynı yapının ve öğrenilmiş olan parametrelerin kullanımına devam edilir. Dağılıma bağımlı olmayan tahmin etme süreçlerinde ise belirlenmiş bir ölçüt (ör: Öklid) ile öğrenme seti yerel parçalara ayrılmakta ve giriş verisi için, kendi alanına denk gelen verilerle eğitilmiş lokal bir model kullanılmaktadır. KA, yapıları gereği dağılımdan bağımsız çalışmaktadır. Çünkü öğrenme süreçlerinin başında, sınıf dağılımları ile ilgili tahminlerde bulunmaz. Karar ağaçlarında, ağacın yapısı baştan belirli olmamaktadır. Veri setinin özelliklerine ve yapısına göre dallar ve yapraklar eklenerek oluşturulmaktadır (Alpaydın, 2017).

Karar ağacı oluşturulması konusunda, C4.5 ve ID3 gibi çeşitli algoritmalar kullanılmaktadır. Bu çalışma kapsamında C4.5 algoritması kullanılmıştır. C4.5, giriş verilerinin sıklıklarına göre sınıflandırma işlemini gerçekleştiren bir algoritmadır. Entropi hesabına dayalı olarak çalışmaktadır. C4.5 çalışma yapısı şu şekilde özetlenebilir (Aksu ve Karaman, 2017):

- (6) nolu eşitlik kullanılarak veri setinin sahip olduğu tüm nitelikler üzerinde entropi hesabı yapılır,
- Her bir nitelik için hesaplanmış olan entropi değeri, sınıfın entropi değerine bölünerek, niteliklerin bilgi kazanç değerleri ortaya çıkartılır,
- En yüksek bilgi kazancı değerine sahip olan nitelik kök olarak seçilir ve dağılım başlar,
- Kök olarak seçilen niteliğin sahip olduğu değerler haricinde kalan diğer nitelikler için aynı işlemler tekrarlanır,
- Tüm veriler işlenip yapraklara ulaşılınca ağaç tamamlanmış olur.

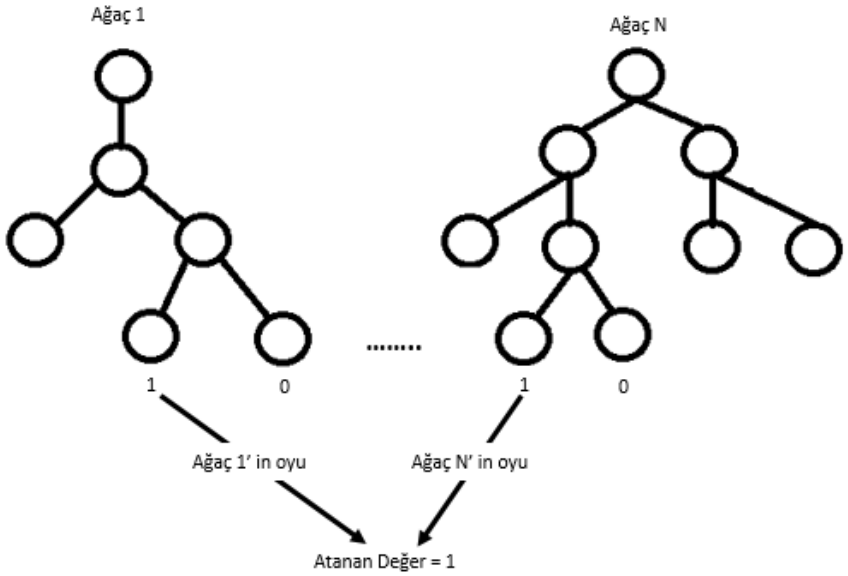
$$Entropi = -\sum_{i=1}^{m} p_i \, log_2(p_i) \quad (6)$$

3.3.6. Rastgele Orman (RO)

Rastgele Orman (RO), birden fazla karar ağacının birleşiminden oluşan bir yapıdır. Orijinal veri içerisinden rastgele çekilmiş birbirlerinden ayrı parçalar kullanılarak, orman içerisinde bulunan ağaçların eğitilmesi esas alınmaktadır. RO içerisinde ağaçların büyümeleri esnasında rastgele özellik seçim işlemi uygulanmaktadır. Bunun da sebebi olarak, çok büyük boyutlardaki veri setleri üzerinden tek bir karar ağacının yeteri kadar sağlıklı sonuç vermesi zor olacağından dolayı, büyük veri setinin parçalara ayrılıp ormandaki karar ağaçlarında ayrı ayrı öğrenme aşamalarının gerçekleştirilmesinin, doğruluk oranlarını arttırması olarak gösterilmektedir (Tanha vd., 2017). Rastgele özellik seçim süreci ile orman içerisindeki ağaçlar, değiştirme yöntemi ile eğitim özelliklerinin alt kümelerinin belirlenmesi ile oluşturulmaktadır. Bu durum, aynı özelliğin birkaç kez seçilebileceği, bazı özelliklerin ise hiç seçilemeyeceği anlamına gelmektedir (Belgiu ve Drăgut, 2016). Bu sayede daha hızlı ve gürültüye daha fazla dayanıklı öğrenme modelleri oluşturulabilmektedir. Buna ek olarak orman içerisindeki ağaçlar**,** modele esneklik kazandırdıkları için, sınıflandırma, kümeleme ve regresyon işlemlerinin performansını arttırmaktadır. Özellikle büyük veri setlerinin değerlendirilmesi aşamasında, iyi bir seçim olarak değerlendirilmektedir (Holzinger vd., 2017).

RO, ağaç yapılı sınıflandırıcıların birleşiminden oluşan bir sınıflandırma algoritmasıdır. Bahsi geçen ağaç yapılı sınıflandırıcılar, birbirlerinden bağımsız ancak, aynı şekilde dağıtılmış rastgele vektörler içermektedir ve yapısı içerisinde bulunan her ağaç, giriş verilerindeki en popüler sınıf değeri için oy verme süreci uygulamaktadır (Breiman, 2001). Ağaçların kullandıkları oylar sonucunda daha fazla ağaçtan oy almış olan sınıf etiketi, RO algoritması tarafından belirlenmiş etiket değeri olarak gösterilir (Belgiu ve Drăgut, 2016). Şekil 8'de RO görsel örneği verilmiştir.

Şekil 8: Rastgele Orman Algoritması, Sınıflandırma Şeması



3.3.7. k - En Yakın Komşu Algoritması (k-NN)

k-NN algoritması, sınıflandırma teknikleri altında benzerlik fonksiyonlarını çalıştıran ve bu şekilde tahmin süreçlerini çalıştıran bir algoritmadır. 1950'li yıllarda keşfedilmiş ve hala günümüzde popüler olarak kullanımı devam etmektedir. Çalışma mantığına, iki boyutlu bir düzlemde bakmak gerekmektedir. Bunun için Şekil 9'da görüldüğü gibi veriler iki boyutlu bir düzleme yerleştirilmekte ve dikey ve yatay eksen (x ve y) değerlerine göre, benzerlik değerleri hesaplanmaktadır (Şeker, 2016).

k-En Yakın Komşu Algoritması (k-NN), makine öğrenmesi algoritmalarının içerisinde en ılımlı çalışan sınıflandırıcı türüdür. k ile ifade edilen benzerlik vektörlerine dayanan çalışma sisteminde, bir nesnenin sınıflandırılması sürecinde k adet komşularının içerisindeki en çok oy alan sınıf değeri kullanılmaktadır. Sınıfı tahmin edilecek olan değerin komşularının adedini belirten k, genellikle çok küçük pozitif bir tamsayıdır (Harefa vd., 2016). k-NN algoritması, sınıf etiketi bilinmeyen yeni bir örnek veri giriş yapıldığında, bu verinin sınıflandırılma sürecinin yakın komşu konumunda bulunan çoğunluk tarafından gerçekleştirilmesi sağlayan denetimli bir algoritma olarak görev yapmaktadır. k-NN ile, test verileri ile algoritmanın sınanması süreçlerinde ve yeni girişi yapılan verinin sınıf etiketinin tespit edilmesi süreçlerinde, bu verilerin öğrenme kümesindeki veriler ile aralarındaki mesafeyi ölçerek, test edilecek veya sınıfı belirlenecek veriye en yakın olan k adet komşuyu bulmaya odaklanmaktadır. Burada bahsedilen mesafenin hesaplanması konusunda da genel anlamda Öklid eşitliğinin (7) kullanılması tercih edilmektedir (Indriani vd., 2017).

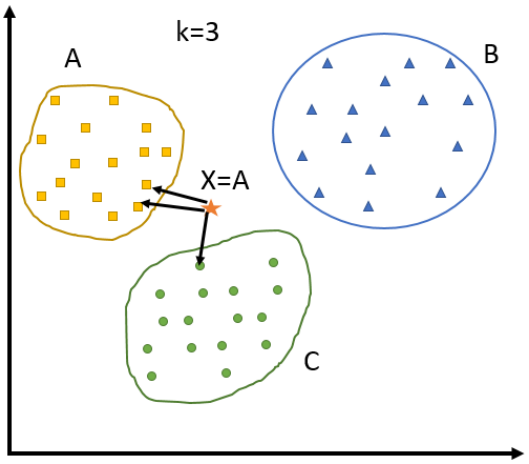
$$E(i,j) = \sqrt{\sum_{k=1}^{n}(i_{k-}\,_{j_k})^2} \tag{7}$$

k-NN, temel manada verilerin belirli bir özellik alanına ait olduğunu varsaymaktadır. Bu sebeple, yeni girişi yapılan veri noktasının, öğrenme kümesindeki tüm noktalara olan uzaklıkları tek tek dikkate alınmaktadır. Yeni girilen verinin sınıf değerinin tespit edilmesi konusunda da uzaklıkları hesaplanmış olan komşular için başta belirlenmiş olan k değeri baz alınır. Eğer k=1 olarak belirlenmişse, sadece en yakın mesafedeki komşunun sınıf değerine göre karar verilir. Uygulamadaki k değerinin optimum seviyede belirlenmesi önemlidir. Çünkü, k çok küçük olursa,

algoritma öğrenme kümesinde bulunan gürültüden çok fazla etkilenecektir. k değeri çok büyük olursa, aslında çok uzak olan komşularda yakın olarak değerlendirilecek ve bunun sonucunda verinin sınıfının tahmininde hata yapma ihtimali artacaktır. k-NN algoritmasının çalışması şu şekilde sıralanabilir (Jadhav ve Channe, 2016):

1. k değeri başlatılır,
2. Giriş verisi ve öğrenme kümesi verileri aralarındaki mesafeler ölçülür,
3. Ölçülen mesafe değerleri sıralanır,
4. k adet en yakın komşular belirlenir,
5. Şekil 9'da ifade edildiği gibi, basit çoğunluk sistemi ile komşular arasında en fazla bulunan sınıf değeri, giriş verisi için tahmin edilir.

Şekil 9: k-NN Algoritması için Örnek Şema (Jadhav ve Channe, 2016).



3.4. Değerlendirme

Çalışma kapsamında kullanılan veri setleri üzerinde makine öğrenmesi algoritmalarının sınanması sürecinde aşağıda belirtilen ölçüm birimleri kullanılmıştır. Bu ölçümlerden elde edilen sonuçlar doğrultusunda, veri setleri üzerinde en uygun değerlemeyi yapan öğrenme algoritmasına karar verilebilmektedir.

3.4.1. Karmaşıklık Matrisi

Karmaşıklık matrisi, öğrenme modellerinin performanslarının görselleştirildiği bir düzen olarak kullanılmaktadır. Herhangi bir algoritma ile sınıflandırma işlemi yapabilmek için oluşturulmuş olan öğrenme modellerinin, gerçek sınıf değerleri bilinen test verileri üzerinde sınanmaları sonucunda, modelin başarımını gösterir. Karmaşıklık matrisi tarafından model performansının belirlenmesi amacıyla yapılan tanımlamalar şu şekildedir (Hasan vd., 2019):

- True Positive (TP): Gerçek Pozitif- Gerçekte 1 olan sınıf etiketlerinin, 1 olarak tahmin edilme sayısı,
- True Negative (TN): Gerçek Negatif- Gerçekte 1 olmayan sınıf etiketlerinin, 1 olarak tahmin edilmemesi,
- False Positive (FP): Yanlış Pozitif- Gerçekte 1 olmayan sınıf etiketlerinin, 1 olarak tahmin edilmesi,
- False Negative (FN): Yanlış Negatif- Gerçekte 1 olan sınıf etiketlerinin, 1 olarak tahmin edilmemesidir.

Tablo 2 ve Tablo 3'te çalışma kapsamında kullanılan veri setlerine ait karmaşıklık matrisleri verilmiştir.

Tablo 2. Standart Veri Seti, Sınıflandırma Algoritmalarına Göre Karmaşıklık Matrisleri

NB	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 16283	FP= 1692
Anomali Var	FN= 4786	TP= 884

YSA	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 17818	FP= 157
Anomali Var	FN= 1527	TP= 4143

LR	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 17947	FP= 28
Anomali Var	FN= 5474	TP= 196

DVM	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 17975	FP= 0
Anomali Var	FN= 5639	TP= 31

C4.5	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 17723	FP= 252
Anomali Var	FN= 403	TP= 5267

RO	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 17871	FP= 104
Anomali Var	FN= 195	TP= 5475

k-NN	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 17834	FP= 141
Anomali Var	FN= 579	TP= 5091

Tablo 3. Optimize Veri Seti, Sınıflandırma Algoritmalarına Göre Karmaşıklık Matrisleri

NB	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 12032	FP= 3085
Anomali Var	FN= 2602	TP= 1915

YSA	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 14944	FP= 173
Anomali Var	FN= 1563	TP= 2954

LR	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 15041	FP= 76
Anomali Var	FN= 4376	TP= 141

DVM	Öngörülen Sınıf Negatif	Öngörülen Sınıf Pozitif
Gerçek Sınıf Negatif	TN= 15114	FP= 3
Gerçek Sınıf Pozitif	FN= 4376	TP= 141

C4.5	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 14894	FP= 223
Anomali Var	FN= 354	TP= 4163

RO	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 14999	FP= 118
Anomali Var	FN= 171	TP= 4346

k-NN	Öngörülen Anomali Yok	Öngörülen Anomali Var
Anomali Yok	TN= 14919	FP= 198
Anomali Var	FN= 553	TP= 3964

3.4.2. Ölçüt İfadeleri

Çalışma kapsamında, karmaşıklık matrisinden elde edilen değerlerin kullanıldığı ve model performansını değerlendirme amacıyla kullanılacak ölçüt ifadeleri ve eşitlikleri şu şekildedir:

- **Model Doğruluğu:** Öğrenme modelinin doğruluk derecesini belirler.

$$Doğruluk = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

- **Model Kesinliği:** Öğrenme modelinin duyarlılık derecesini ölçer.

$$Kesinlik = \frac{TP}{TP+FP} \quad (9)$$

- **Modelin Duyarlılığı:** Test sonucunda gerçek pozitif değerlerin oranı olarak bilinmektedir.

$$Duyarlılık = \frac{TP}{TP+FN} \quad (10)$$

- **F Ölçütü:** Duyarlılık ve Kesinlik değerlerinin harmonik ortalamasıdır.

$$F\ Ölçütü = \frac{2*Kesinlik*Duyarlılık}{Kesinlik+Duyarlılık} \quad (11)$$

- **Hata Oranı:** Hatalı ölçümlerin, toplam değer sayısına olan oranının ölçümüdür.

$$Hata\ Oranı = \frac{FP+FN}{TP+TN+FP+FN} \quad (12)$$

3.5. Uygulama

Çalışmada belirtilen veri setleri üzerinde, ilgili algoritmaların uygulanması, öğrenme modellerinin oluşturulması ve sonuçların elde edilmesi süreçlerinde, Intel Core i5-3230M model, 2.6 Ghz frekansa sahip çift çekirdekli CPU, 6 Gbyte 1600 Mhz RAM bellek, Intel HD Graphics 4000 paylaşımlı ekran kartı özelliklerine sahip bir dizüstü bilgisayar kullanılmıştır. Öğrenme modellerinin oluşturulmasında, açık kaynak kodlu Weka yazılımı kullanılmıştır.

3.5.1. Ölçümlerin Analizi

Tablo 4, Tablo 5 ve Şekil 10' da gösterilen sonuçlar incelendiğinde, Yapay Sinir Ağları, C4.5 Karar Ağacı, Rastgele Orman ve k En Yakın Komşu algoritmaları tarafından oluşturulan öğrenme modellerinin yüksek başarı gösterdiği görülmektedir. Ayrıca bu modeller içerisinden, standart veri setinin test kümesinde bulunan 1148 adet anomali verisinin 1099 adedini, optimize edilmiş veri setinin test kümesinde bulunan 938 adet anomali verisinin 891 adedini tahmin ederek, en yüksek doğruluk oranı ile en iyi tahmin modelinin Rastgele Orman algoritması tarafından oluşturulduğunu söyleyebiliriz.

Bunların dışında kalan Naive Bayes, Lojistik Regresyon ve Destek Vektör Makineleri algoritmaları ise başarılı tahmin konusunda önemli ölçüde geride kalmış durumdadırlar. Özellikle her iki veri seti için de gerçekte anomali olan çok yüksek miktarda verinin, bu algoritmalar tarafından normal veri olarak tahmin edildiği görülmüştür. Sonuç olarak adı geçen bu algoritmalar, veri setlerinde belirtilmiş olan anomali verilerinin tespit edilmesi konusunda büyük oranda başarısız olmuşlardır.

Ayrıca algoritmalar üzerinde yapılan karşılaştırmalar sonucunda, standart çalışma düzeni ile oluşturulmuş olan veri seti ile optimize edilmiş çalışma düzeni ile oluşturulmuş olan veri setinin ölçümlerinde kayda değer bir fark gözlenmemiştir.

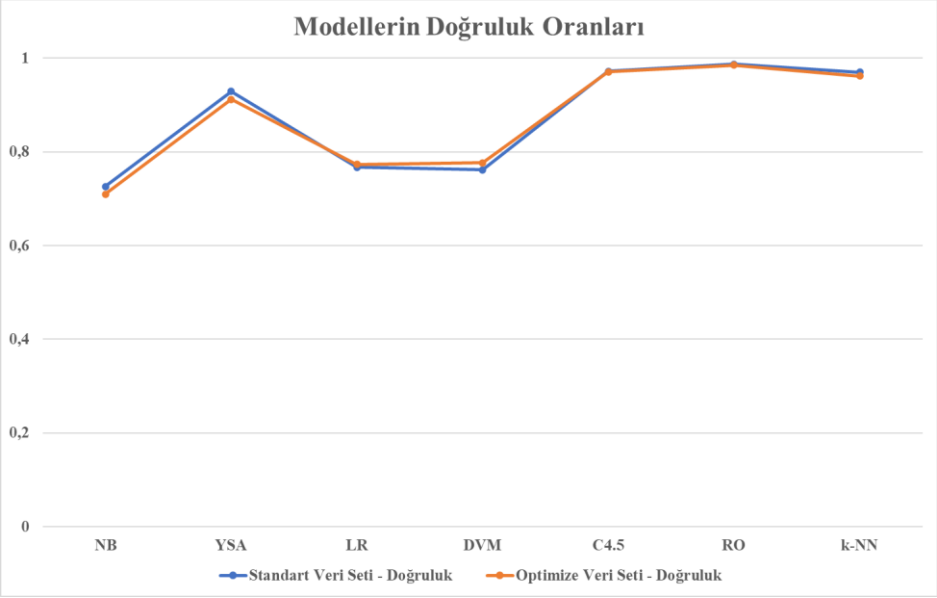
Tablo 4. Standart Veri Seti, Uygulama Sonuçları

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F-Ölçümü**	**Hata Oranı**
NB	0,726	0,343	0,156	0,214	0,274
YSA	0,929	0,963	0,731	0,831	0,071
LR	0,767	0,875	0,035	0,067	0,233
DVM	0,762	1,000	0,005	0,011	0,238
C4.5	0,972	0,954	0,929	0,941	0,028
RO	0,987	0,981	0,966	0,973	0,013
k-NN	0,970	0,973	0,898	0,934	0,030

Tablo 5. Optimize Veri Seti, Uygulama Sonuçları

	Doğruluk	**Kesinlik**	**Duyarlılık**	**F-Ölçümü**	**Hata Oranı**
NB	0,710	0,383	0,424	0,402	0,290
YSA	0,912	0,945	0,654	0,773	0,088
LR	0,773	0,650	0,031	0,060	0,227
DVM	0,777	0,979	0,031	0,061	0,223
C4.5	0,971	0,949	0,922	0,935	0,029
RO	0,985	0,974	0,962	0,968	0,015
k-NN	0,962	0,952	0,878	0,913	0,038

Şekil 10: Öğrenme Modellerinin, Standart Veri Seti ve Optimize Veri Seti Üzerindeki Doğruluk Oranları Karşılaştırması



4. TARTIŞMA VE SONUÇ

Çalışma kapsamında, kaggle ortamında bulunan, enerji optimizasyonu üzerine hazırlanmış bir konveyör bant sistemi tarafından oluşturulmuş, çeşitli anomaliler içeren iki farklı açık kaynak veri seti temin edilerek kullanılmıştır. Birinci veri setinde, depolama sisteminin ortasındaki iki bant dikey yönlü hareket esnasında yatay yönlü hareket sergilemeden çalışmış ve bu şekilde enerji optimizasyonu değerlendirilmiştir. İkinci veri setinde ise ortadaki iki bant dikey ve yatay hareketleri aynı anda gerçekleştirmesi ile enerji optimizasyonu değerlendirilmiştir. Temin edilen bu veri setleri üzerinde, Naive Bayes, Yapay Sinir Ağları, Lojistik Regresyon, Destek Vektör Makineleri, Karar Ağacı, Rastgele Orman ve k En Yakın Komşu isimli sınıflandırma algoritmaları kullanılarak öğrenme modelleri oluşturuşmuş ve veri setlerinde var olan anomali verilerinin tespit edilmesi noktasında öğrenme modelleri test edilerek karşılaştırılmıştır.

Ayrıca ilgili veri setleri, Hranisavljevic ve ekibi tarafından önerilen Derin Ağ Zamanlı Otomat (DENTA) isimli bir modelin testinde kullanılmış ve sonucunda özellikle gerçek dünya veri kümeleri üzerinde anomali tespiti konusunda avantajlı sonuçların elde edildiği bildirilmiştir (Hranisavljevic vd., 2020). Kim ve ekibi tarafından önerilen Projeksiyon Yolu Boyunca Yeniden Yapılanma (RaPP) isimli yenilik tespit sistemi modeli kapsamında oluşturdukları otomatik kod çözücülerin (AE) değerlendirilmesinde ise çok çeşitli özelliklere sahip veri setleri sınanmıştır. Bu sınama sürecinde kullanılan veri setlerinden bir tanesi de bu çalışmada analiz edilen yüksek raflı depolama sistemlerinin enerji optimizasyonu için oluşturulmuş olan veri kümesidir. Genel anlamda önerilen modelin analizler sonucunda iyi bir performans gösterdiği ifade edilmiştir (Kim vd, 2019). Shin ve Kim tarafından yapılan çalışmada, yine RaPP için geliştirilen genişletilmiş otomatik kod çözücünün (XAE) performansının analiz edilmesi amacıyla, aynı veri setleri kullanılmış ve genel manada önceki çalışmaya göre bir miktar daha iyi doğruluk sonuçlarına erişildiği gösterilmiştir (Shin ve Kim, 2020). Bahsi geçen çalışmalar ile bu çalışmada elde edilen en yüksek doğruluk oranlarına sahip Rastgele Orman algoritmasının değerlerinin karşılaştırmaları Tablo 6'da verilmiştir.

Tablo 6. Farklı Çalışmalara Ait Sonuçların Karşılaştırılması

	RO (Standart Veri Seti)	**RO (Optimize Veri Seti)**	**DENTA**	**RaPP - AE**	**RaPP - XAE**
Doğruluk Oranı	0,987	0,985	0,812	0,650	0,631

Sonuç olarak, çalışma içerisinde yaptığımız analizler sonucunda, Yapay Sinir Ağları, Karar Ağacı, Rastgele Orman ve k En Yakın Komşu algoritmaları tarafından oluşturulmuş olan öğrenme modelleri anomali tespiti konusunda başarı elde etmiştir. Ayrıca bu algoritmalar içerisinde doğruluk oranı ve F-Ölçümü değeri ile Rastgele Orman algoritmasının öğrenme modeli hem bu çalışmada, hem de diğer çalışmalarda uygulanan modeller arasında en iyi anomali tespiti yapan model olmuştur.

Çalışmanın bundan sonraki süreçlerinde, zeki fabrikalar içerisinde çalışan siber fiziksel sistemler üzerinde, özellikle siber saldırılar tarafından meydana gelebilecek anomalilerin anlık olarak tespit edilmesi ile ilgili çalışmaların yapılması planlanmaktadır.

KAYNAKÇA

Akçetin, E., & Çelik, U. (2014). İstenmeyen Elektronik Posta (Spam) Tespitinde Karar Ağacı Algoritmalarının Performans Kıyaslaması. *İnternet Uygulamaları ve Yönetimi, 5*(2), 43-56.

Aksu, M. Ç., & Karaman, E. (2017). Karar Ağaçları ile Bir Web Sitesinde Link Analizi ve Tespiti. *ACTA INFOLOGICA, 1*(2), 84-91.

Alpaydın, E. (2017). *Yapay Öğrenme.* İstanbul: Boğaziçi Üniversitesi Yayınevi.

Arı, A., & Berberler, M. E. (2017). Yapay Sinir Ağları ile Tahmin ve Sınıflandırma Problemlerinin Çözümü İçin Arayüz Tasarımı. *Acta - Infologica, 1*(2), 55-73.

Aydemir, E. (2019). *Weka ile Yapay Zeka.* Ankara: Seçkin Yayıncılık.

Bagozi, A., Bianchini, D., Antonellis, V. D., Marini, A., & Ragazzi, D. (2017). Big Data Summarisation and Relevance Evaluation for Anomaly Detection in Cyber Physical Systems. *OTM 2017: On the Move to Meaningful Internet Systems* (s. 429-447). Rhodes, Greece: Springer.

Belgiu, M., & Drăgut, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*(114), 24-31.

Birgelen, A. v., & Niggeman, O. (2017). Using self-organizing maps to learn hybrid timed automata in absence of discrete events. *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (s. 1-8). Limassol: IEEE.

Breiman, L. (2001). Random Forests. *Machine Learning*(45), 5-32.

Brownlee, J. (2019). *Machine Learning Mastery With Weka.*

Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2017). Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access*(6), 6505-6519.

Elmas, Ç. (2016). *Yapay Zeka Uygulamaları 3. Baskı.* Ankara: Seçkin Yayıncılık.

Feizizadeh, B., Roodposhti, M. S., Blaschke, T., & Aryal, J. (2017). Comparing GIS-based support vector machine kernel functions for landslide susceptibility mapping. *Arabian Journal of Geosciences, 10*(117).

Frank, A. G., Dalenogare, L. S., & Ayala, N. F. (2019). Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*(210), 15-26.

Gürsakal, N. (2018). *Makine Öğrenmesi.* Bursa: Dora Yayınevi.

Han, J., & Kamber, M. (2006). *Data Mining, Concepts and Techniques 2nd Edition.* San Francisco: Morgan Kaufmann Publishers.

Hand, D., Manila, H., & Smyth, P. (2001). *Principles of Data Mining.* London: Massachusetts Institute of Technology.

Harefa, J., Alexander, A., & Pratiwi, M. (2016). Comparison Classifier: Support Vector Machine (SVM) and K-Nearest Neighbor (K-NN) In Digital Mammogram Images. *Jurnal Informatika dan Sistem Informasi, 2*(2), 35-40.

Hasan, M., Islam, M. M., Zarif, M. I., & Hashem, M. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, 1-14.

Holzinger, A., Malle, B., Kieseberg, P., Roth, P. M., Müller, H., Reihs, R., & Zatloukal, K. (2017). Machine Learning and Knowledge Extraction in Digital Pathology Needs an Integrative Approach. A. Holzinger, R. Goebel, M. Ferri, & V. Palade içinde, *Towards Integrative Machine Learning and Knowledge Extraction* (s. 13-50). Springer.

Hranisavljevic, N., Maier, A., & Niggeman, O. (2020). Discretization of hybrid CPPS data into timed automaton using restricted Boltzmann machines. *Engineering Applications of Artificial Intelligence*(95), 1-9.

Hranisavljevic, N., Niggemann, O., & Maier, A. (2016). A Novel Anomaly Detection Algorithm for Hybrid Production Systems based on Deep Learning and Timed Automata. *The 27th International Workshop on Principles of Diagnosis: DX*. Denver, USA.

Hranisavljevic, N., Niggemann, O., & Maier, A. (2018, 07 19). *High Storage System Data for Energy Optimization*. 03 15, 2020 tarihinde Kaggle: https://www.kaggle.com/inIT-OWL/high-storage-system-data-for-energy-optimization adresinden alındı

Hsieh, R.-J., Chou, J., & Ho, C.-H. (2019). Unsupervised Online Anomaly Detection on Multivariate Sensing Time Series Data for Smart Manufacturing. *IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)* (s. 90-97). Kaohsiung, Taiwan: IEEE.

Indriani, O. R., Kusuma, E. J., Sari, C. A., Rachmawanto, E. H., & Setiadi, D. I. (2017). Tomatoes classification using K-NN based on GLCM and HSV color space. *International Conference on Innovative and Creative Information Technology (ICITech)* (s. 1-6). Salatiga: IEEE.

İşçimen, B., Kutlu, Y., Reyhaniye, A. N., & Turan, C. (2014). Balık tanınmasında görüntü analiz yöntemleri. *22nd Signal Processing and Communications Applications Conference.* Trabzon.

Jadhav, S. D., & Channe, S. P. (2016). Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques. *International Journal of Science and Research*, *5*(1), 1842-1845.

Jain, M., Narayan, S., Pratibha, B., Bhowmick, A., & Muthu, R. (2018). Speech Emotion Recognition using Support Vector Machine. *International Conference on Informatics Computing in Engineering Systems (ICICES).* Chennai, India: IEEE.

Kim, K. H., Shim, S., Lim, Y., Jeon, J., Choi, J., Kim, B., & Yoon, A. S. (2019). RaPP: Novelty Detection with Reconstruction along Projection Pathway. *International Conference on Learning Representations (ICLR 2020),* (s. 1-10). Addis Ababa, Ethiopia.

Mansournia, M. A., Geroldinger, A., Greenland, S., & Heinze, G. (2018). Separation in Logistic Regression: Causes, Consequences, and Control. *American Journal of Epidemiology, 187*(4), 864-870.

Nizam, H., & Akın, S. S. (2014). Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Sosyal Medyada Makine Öğrenmesi ile Duygu Analizinde Karşılaştırılması. *XIX. Türkiye'de İnternet Konferansı.* İzmir.

Pahl, M.-O., & Aubet, F.-X. (2018). All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection. *14th International Conference on Network and Service Management (CNSM 2018)* (s. 72-80). Italy: Aconf.

Patil, T. R., & Sherekar, S. S. (2013). Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification. *International Journal Of Computer Science And Applications, 6*(2), 256-261.

Radziwon, A., Bilberg, A., Bogers, M., & Madsen, E. S. (2014). The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions. *Procedia Engineering*(69), 1184-1190.

Riordan, A. O., Coady, J., Toal, D., Newe, T., & Dooly, G. (2019). Industry 4.0: Pillars for Smart Manufacturing - A Review. *no. February*.

Shin, S. Y., & Kim, H.-J. (2020). Extended Autoencoder for Novelty Detection with Reconstruction along Projection Pathway. *Applied Sciences, 10*(13), 1-14.

Staub, S., Karaman, E., Kaya, S., Karapınar, H., & Güven, E. (2015). Artificial Neural Network and Agility. *Procedia - Social and Behavioral Sciences, 195*, 1477-1485.

Şeker, H. İ., Tuna, M., & Koyuncu, İ. (2018). Gerçek Zamanlı Wavelet Dönüşümleri için FPGA-Tabanlı Meksika Şapkası Dalgacığının Tasarımı ve Gerçeklenmesi. *3rd International Conference on Engineering Technology and Applied Sciences (ICETAS)* , (s. 168-173). Skopje Macedonia.

Şeker, Ş. E. (2016). *Weka ile Veri Madenciliği.* İstanbul: Bilgisayar Kavramları Yayınları.

Tanha, J., Someren, M. V., & Afsarmanesh, H. (2017). Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*(8), 355-370.

Wan, J., Li, J., Imran, M., Li, D., & Amin, F.-e. (2019). A Blockchain-Based Solution for Enhancing Security and Privacy in Smart Factory. *IEEE Transactions on Industrial Informatics, 15*(6), 3652-3660.

Wang, R., Nie, K., Wang, T., Yang, Y., & Long, B. (2020). Deep Learning for Anomaly Detection. *13th International Conference on Web Search and Data Mining* (s. 894-896). Houston, TX, USA: WSDM.

Yakut, E., Elmas, B., & Yavuz, S. (2014). Yapay Sinir Ağları ve Destek Vektör Makineleri Yöntemleriyle Borsa Endeks Tahmini. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 19*(1), 139-157.

Yoon, S., Um, J., Suh, S.-H., Stroud, I., & Yoon, J.-S. (2019). Smart Factory Information Service Bus (SIBUS) for manufacturing application: requirement, architecture and implementation. *Journal of Intelligent Manufacturing*(30), 363-382.