T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

UÇAK MOTORLARI BAKIM ONARIM SÜREÇLERİ İÇİN OPTİMAL KARAR DESTEK SİSTEMLERİ GELİŞTİRİLMESİ

Lilya UMEROVA

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Danışman

Öğr. Gör. Dr. Ömer Özgür BOZKURT

Temmuz, 2019

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

UÇAK MOTORLARI BAKIM ONARIM SÜREÇLERİ İÇİN OPTİMAL KARAR DESTEK SİSTEMLERİ GELİŞTİRİLMESİ

Lilya UMEROVA tarafından hazırlanan tez çalışması 19.07.2019 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Öğr. Gör. Dr. Ömer Özgür BOZKURT

Yıldız Teknik Üniversitesi

Danışman

Jüri Üyeleri

Öğr. Gör. Dr. Ömer Özgür BOZKURT, Danışman
Yıldız Teknik Üniversitesi _____

Doç. Dr. Sırma YAVUZ, Üye
Yıldız Teknik Üniversitesi _____

Dr. Öğr. Üyesi Oğuz ATA, Üye
Altınbaş Üniversitesi _____

Danışmanım Öğr. Gör. Dr. Ömer Özgür BOZKURT sorumluluğunda tarafımca hazırlanan çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Lilya UMEROVA

İmza

Değerli aileme

ve

hocama

TEŞEKKÜR

Yüksek lisans derslerini tamamladıktan sonra bana öncülük eden, moral desteği veren ve ilgisini hiç esirgemeyen, araştırmanın nasıl yapılacağını, bilimsel kaynakların nasıl taranacağını öğreten, yılmadan, sıkılmadan azimle nasıl çalışılacağını gösteren ve Yüksek Lisans Tez Danışmanlığımı yaparken gerekli zamanı ayırarak doğru konuları araştırmamı sağlayan Öğr. Gör. Dr. Ömer Özgür BOZKURT'a içten minnet ve şükranlarımı sunarım. Tüm yaşamım boyunca en büyük destekçim olan ve her türlü maddi, manevi desteğini benden hiçbir zaman esirgemeyen aileme en içten saygı, sevgi ve şükranlarımı sunarım.

Lilya UMEROVA

İÇİNDEKİLER

SİMGE LİSTESİ viii
KISALTMA LİSTESİ ix
ŞEKİL LİSTESİ x
TABLO LİSTESİ xii
ÖZET xiii
ABSTRACT xv
1 Giriş 1
1.1 Literatür Özeti 1
1.2 Tezin Amacı 10
1.3 Hipotez 14
2 Materyal ve Yöntem 15
2.1 KDS Tanıtımı ve İncelenmesi 15
2.1.1 KDS Mimarisi 15
2.2 KDS Alt Sistemlerin Tanıtımı 23
2.3 KDS Özellikleri 24
2.4 KDS Sınıflandırılması 24
2.4.1 KDS Çeşitliği 27
2.5 Uzman Sistemler 31
2.6 KDS Geliştiriken Kullanılan Yöntemler 32
2.6.1 Karar Ağaçları 32
2.6.2 Genetik algoritmalar 33
2.6.3 Bulanık Mantık 34
2.6.4 YSA 37

3 Deneysel Bulgular **43**

3.1 KDS Tasarımı ... 45

3.1.1 KDS Yapısının Geliştirilmesi ... 45

3.1.2 KDS Diyagramı ... 47

3.2 Motor Bakım Kavramları ... 49

3.2.1 Motor Basınç Oranı ... 50

3.2.2 N1 Hız ... 50

3.2.3 Egzoz Gazı Sıcaklığı(EGS) ... 50

3.2.4 EGS Marjı ... 50

3.2.5 Motorun Sınırlı Ömür Parçaları ... 50

3.2.6 Motorun İtme Kuvveti ... 52

3.2.7 Mekanik Hasarlar ... 53

3.3 Motor Bakımı ... 53

3.4 Maliyet ... 53

3.5 Optimizasyon Kuralları ... 55

3.5.1 Optimizasyon Kuraların belirlenmesi Genetik ... 55

3.6 Optimizasyon Parametreleri ... 60

3.6.1 Veri Toplama ve Normalizasyon ... 60

3.6.2 Optimizasyon Yöntemlerin Analizi ... 61

3.7 Sistemin Tanıtılması ... 64

3.7.1 KDS Çözüm Algoritması ... 65

3.7.2 Uygulama Modeli ... 65

3.7.3 Basamaklandırma İşlemin Sonucu ... 75

3.7.4 Uygulama ... 78

4 Sonuç ve Öneriler **87**

A Ekler **89**

Kaynakça **98**

SİMGE LİSTESİ

X	Evrensel küme.
μ	Üyelik derecesi.
μA(x)	x'in A bulanık kümesindeki değeri.
[0,1]	0 ve 1 arasında sonsuz değer aralığı.
{}	Küme.
Ø	Boş Küme
Σ	Toplam sembolü
z*	Berraklaştırma işlemi sonucu
×	Kartezyen çarpımı
%	Yüzde

KISALTMA LİSTESİ

BÇS	Bulanık Çıkarım Sistemi
BT	Bilgi Teknolojileri
CSN	Motorun İmal tarihinden İtibaren Çalıştırma Sayısı
CSO	Motorun Son Büyük Bakımdan Sonraki Çalıştırma Sayısı
EASA	Avrupa Havacılık Güvenliği Ajansı
ERP	Kurumsal Kaynak Planlama (Enterprise Resource Planning)
FAA	Federal Havacılık İdaresi
FC	Motor Çalıştırma Sayısı (Flight Cycle)
GKDS	Grup Karar Destek Sistemi
KDS	Karar Destek Sistemi
MM	Bakım El Kitabı (Sivil) (Maintenance Manual)
OKDS	Organizasyonel Karar Destek Sistemi
SHGM	Sivil Havacılık Genel Müdürlüğü
SQL	Yapısal Sorgu Dili (Structured Query Language)
TSN	Uçağın İmal Tarihinden İtibaren Uçtuğu Toplam Uçuş Saati
TSO	Uçağın Son Büyük Bakımdan Sonraki Uçuş Saati
YBS	Yönetim Bilgi Sistemi
YSA	Yapay Sinir Ağları

ŞEKİL LİSTESİ

Şekil 2.1 Fonksiyonel KDS 16
Şekil 2.2 Bağımsız Veri Marketleri 17
Şekil 2.3 İki Katmanlı Veri Ambarı 18
Şekil 2.4 Üç Katmanlı Veri Ambarı 19
Şekil 2.5 KDS'in İlgili Olduğu Bilim Dalları [80] 21
Şekil 2.6 KDS Alt Sistemleri [81] 23
Şekil 2.7 Bulanık Kontrol Sisteminin Yapısı 36
Şekil 2.8 Tek Katmanlı Algılayıcı 39
Şekil 2.9 Dört Nöronlu Hopfield Ağı 39
Şekil 2.10 Üç Giriş ve Beş küme Elemanına Sahip Kohonen Ağı 40
Şekil 3.1 KDS Ana Aşamaları 46
Şekil 3.2 KDS'nin Yapısı 47
Şekil 3.3 KDS'nin Şeması (a) 48
Şekil 3.4 KDS'nin Şeması (b) 49
Şekil 3.5 Kalan Ömür Hesabı 51
Şekil 3.6 Kalan Ömür Takibi 52
Şekil 3.7 Rapor Örneği 52
Şekil 3.8 Basamaklandırma, Motor Bakım/Söküm Hesabı 56
Şekil 3.9 Basamaklandırma 57
Şekil 3.10 KDS'nin Havayolu Veri Tabanıyla Entegrasyon Şeması 61
Şekil 3.11 Bulanık Eşleme Yaklaşımı 68
Şekil 3.12 Nöro-bulanık Sistemin Yapısı 69
Şekil 3.13 Nöro-bulanık Hibrit Ağın Genel Yapısı 69
Şekil 3.14 YSA Modeli 70
Şekil 3.15 Üyelik Fonksiyonları 71
Şekil 3.16 LLP Ömür Kriteri için Üyelik Fonksiyonu 71
Şekil 3.17 Gerçek Motor Söküm Planı 78
Şekil 3.18 Hesaplanan Motor Söküm Planı 78
Şekil 3.19 Ağ işleyişi 79

Şekil 3.20 Örnek Parametre Değerleri Matrisi 80
Şekil 3.21 Ağırlık Matrisi 80
Şekil 3.22 Hesaplanan Ağırlık Matrisi 81
Şekil 3.23 Hata Değerleri 81
Şekil 3.24 Bulanık Mantık Örneği 85

TABLO LİSTESİ

Tablo 2.1 KDS Mimarisi 20
Tablo 2.2 KDS Değerlendirmesi 21
Tablo 2.3 KDS Gereksinimleri 30
Tablo 2.4 Veri Madenciliği Yöntemleri ve Araçları 30
Tablo 2.5 KDS Tasarım Yöntemleri 41
Tablo 3.1 Motor Değişim Senaryosu 1 59
Tablo 3.2 Motor Değişim Senaryosu 2 59
Tablo 3.3 Optimizasyon Yöntemlerin Öğrenme Değerleri 64
Tablo 3.4 Motor Kriterlerin Ölçüm Değerleri 72
Tablo 3.5 Kriter Ağırlıkları 72
Tablo 3.6 Kriter Matrisi 73
Tablo 3.7 KDS'nin Hesapladığı Bakım/Söküm Tarihleri 73
Tablo 3.8 KDS Sonuç Karşılaştırılması 74
Tablo 3.9 Sıralı Sonuç Değerleri 75
Tablo 3.10 Örnek Senaryo 1 76
Tablo 3.11 Örnek Senaryo 2 77
Tablo 3.12 Parametreleri Sınır Değerleri 79
Tablo 3.13 Dilsel Değişkenler 84
Tablo 3.14 YSA Başarı Ölçümü 86

ÖZET

UÇAK MOTORLARI BAKIM ONARIM SÜREÇLERİ İÇİN OPTİMAL KARAR DESTEK SİSTEMLERİ GELİŞTİRİLMESİ

Lilya UMEROVA

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Öğr. Gör. Dr. Ömer Özgür BOZKURT

Bu tezde, havacılık sektöründe karmaşık karar problemlerinin optimal olarak çözülebilmesi için karar destek sistemlerinin geliştirilmesi araştırılmıştır. Karar problemleri genel olarak birden fazla kriter ve çoğu zaman belirsiz ve doğrusal olmayan özellikler içermektedir. Bu problemleri çözebilmek için bulanık mantık ve yapay sinir ağları gibi yapay zekâ tekniklerine ihtiyaç duyulmaktadır. Belirsizlik seviyesi yüksek bu tür problemleri çözmek için analitik modeller kullanan karar destek sistemlerinden faydalanılabilmektedir.

Çalışmada; uçak motorlarının bakım, onarım ve yenileme süreçlerindeki projelerde hangi işe ne zaman başlanıp ne zaman bitirilmesi gerektiği belirleyecek ve kısıtlı sayıdaki kaynakların (parçalar, takımlar, bakım kartları, söküm takım işlemleri) optimal atanmasını sağlayacak bir karar destek sistemin geliştirilmesi ve gerçekleşmesinin yanı sıra, projenin hedeflendiği tarihte tamamlanmasına destek olarak tüm kaynakların en verimli şekilde kullanılmasına yönelik bir optimizasyon yöntemi geliştirilmesi sağlanmıştır. Tez çalışmasında sonuç olarak, havayolların

düşük maliyetli bakım planlarının oluşturulabileceğine ve maliyeti düşük düzeyde tutabileceğine yönelik tavsiyelerde bulunulmuştur.

Anahtar Kelimeler: Karar destek sistemi, yapay sinir ağları, bulanık mantık

ABSTRACT

Development of Optimal Decision Support System for the Maintenance Processes of Aircraft Engines.

Lilya UMEROVA

Department of Computer Engineering

Master's Thesis

Advisor: Inst. Dr. Ömer Özgür BOZKURT

In this thesis, the development of decision support systems was investigated in order to optimally solve complex decision problems in the aviation sector. Decision problems generally contain more than one criterion and often contain ambiguous and non-linear features. In order to solve these problems is a need for artificial intelligence techniques such as fuzzy logic and neural networks. Decision support systems (DSS) provide decision makers with analytical models to solve such decision problems with high levels of uncertainty.

In this study; Development and implementation of the Decision support system to ensure the optimal assignment of a limited number of resources (parts, tools, maintenance cards, dismantling operations) in the work of aircraft engines during the maintenance, repair and renovation processes. Development of a solution that will ensure the completion of the project at the targeted date and thus the optimization method for the most efficient use of all resources will be

implemented to determine the solution. As a result of the thesis, it was suggested that airlines could create low-cost maintenance plans and keep costs low.

Keywords: Decision support systems, neural network, fuzzy logic

1
Giriş

1.1 Literatür Özeti

Karar verme, önemli yaşam becerilerinden biridir. Karar verme birden fazla seçeneğin bulunması durumunda bunlar arasından bir seçim yapılmasıdır. Bir problemin varlığını algılamak, değişik kaynaklardan bilgiler toplayarak problemi tanımlamak, söz konusu problemin değişik çözüm yollarını ortaya koymak, bunları birbirleri ile karşılaştırarak en faydalı olanını seçmek ve böylece bir uygulama planı elde etmektir [1].

Tüm şirketler ve işletmeler, risk ve belirsizlik koşullarında faaliyet gösterdiğinden, hayatta kalmanın yanı sıra operasyonun sürekliliğini ve gelişimini sağlama sorununun üstesinden gelme çabasındadır. Büyük işletmelerin işleyişlerinin analizi ve yönetimi, bilgi işlem alanındaki yönetici, uzman ve analistlerin ana çalışma konusudur. Bu sorunu çözmek için yöneticiler, işletmenin gelişim yönünü seçme, pazar konumlarını ve pazar paylarını koruma ve arttırma, kayıpları önleme ve maliyetleri azaltma, rekabet gücünü koruma ve geliştirme konularında yönetim kararları almak zorundadır. Alınan kararların sağlamlığı ve profesyonel karar verme düzeyi nihayetinde işletmenin verimliliğini belirler [2, 6].

Modern koşullarda, bilgi önemli bir rol oynamaktadır. Ancak yönetim kararlarını verirken göz önünde bulundurulması gereken çok sayıda ekonomik, politik, sosyal, yasal faktörler doğru çözümü seçme sürecini önemli ölçüde zorlaştırmaktadır. Genellikle, konuyla ilgili, güvenilir, sağlam ve eksiksiz bilgi toplama sürecinde zorluklar yer almaktadır. Gelen ve işlenen bilginin büyüklüğündeki hızlı artış, bilgiyi analiz etme yönteminde önemli değişikliklere neden olur. Verilerin işlenmesi ve incelenmesi, aynı zamanda bilginin ve süreçlerin entelektüelleştirilmesi, etkin yöntemlerin ve akıllı karar destek teknolojilerinin inşa edilmesi ve uygulanmasını gerektirir.

Son yıllarda, bilgi teknolojileri yüksek bir gelişim seviyesine ulaşmıştır. Bu bağlamda, çoğu gelişmekte olan şirket, biriken verilerini verimli bir şekilde depolamak, işlemek ve dağıtmak için otomatik araçlar kullanmaktadır [3]. Yönetim bilgi sistemlerinin (YBS) ve veri tabanı yönetim sistemlerinin geliştirilmesi sonucunda modern karar destek sistemleri (KDS) ortaya çıkmıştır, KDS yönetim faaliyetlerinde ortaya çıkan sorunları çözmek için uyarlanmış sistem olarak da tanımlanabilir.

1980 yılında Steven Alter doktora tezi çalışmasının sonuçlarını "Decision Support Systems: Current Practice and Continuing Challenge" isimli bir kitap olarak yayınlamıştır. Bu çalışma KDS altyapısına ve uygulamalarına büyük katkılar sağlamıştır. 1982'de Ralph Sprague ve Eric Carlson' un "Building Effective Decision Support Systems" isimli kitabı KDS için önemli bir eserdir. Bu kaynak, örgütlerin KDS'yi neden kurması gerektiği ve nasıl kurabileceğine ilişkin pratik ve anlaşılır bir bakış açısı sağlamıştır [5].

Sprague 1980 yılında yayınlamış olan [4] çalışmasında KDS'yi aşağıdaki gibi tanımlamaktadır:

1. KDS, üst düzey yöneticilerin genellikle karşılaştığı daha az yapılandırılmış, belirsiz olmayan problemi hedef alır.
2. KDS, modellerin veya analitik tekniklerin kullanımını geleneksel veri erişimi ve karar verme işlevleriyle birleştirmeye çalışır.
3. KDS özellikle, bilgisayarla yetkin olmayan kişiler tarafından etkileşimli bir biçimde kullanılmasını kolaylaştıran özelliklere odaklanır.
4. KDS, ortamdaki değişiklikleri ve kullanıcının karar verme yaklaşımını uyarlamak için esneklik ve uyarlanabilirlik sağlar.

Karar verme analizi 1950 – 1960 yıllar arası Carnegie Institute Of Technology tarafından yapılmıştır [3]. Massachussets Institute of Technology (MIT) 1960 yılında bilgisayar tabanlı karar verme teorisini kullanmaya başlanmıştır. MIT tarafından geliştirilen ve karmaşık dağıtım zinciri yönetimi ilkelerini öğretmek için kullanılan "Beer Game" günümüzde tek kişilik PC uyarlaması ve internet üzerinden birden fazla kişinin oynayabildiği versiyonu ile etkileşimsel ve aktif eğitim

metotlarının sayısal yöntemler konusunda sınıf ortamından elektronik ortama geçmesinin en iyi örneğidir [73]. 1980'lerde KDS için yoğun ilişkilerinden, Organizasyonel Karar Destek Sistemi (OKDS), Grup Karar Destek Sistemi (GKDS) ve Yönetim Bilgi Sistemi (YBS) gibi tanımlar KDS başlığı altında toplanmıştır. 1990 yılında ise veri ambarı ve analitik çözümü kapsayacak şekilde genişletilmiştir [3].

KDS karar vericilerin karmaşık, yapılandırılmamış görevleri/problemleri çözmelerine yardımcı olan güçlü bir araçtır. Karar destek sistemleri veri tabanı teorisi, benzetim, modelleme yöntemleri, yapay zekâ, sinir ağları, durum analizi ve etkileşimli bilgisayar sistemlerini içeren çok disiplinli bir çalışmanın sonucudur. Literatürde KDS için farklı tanımlar yer almaktadır, bunlardan bazıları aşağıda verilmiştir:

- Karar destek sistemleri plansız karar vermeyi destekleyen özelleştirilmiş yönetim sistemleridir [7].
- Karar destek sistemleri karar vericinin analiz kabiliyetlerini artırıcı bilgi sunarak onları desteklemeyi, fırsatlar konusunda uyarılar yaparak tavsiyelerde bulunmayı ve yapısal olmayan kararlarda yardım sunarak problem çözümünü kolaylaştırmayı amaçlar [8].
- Sistem, yöneticisine karar vermesine yardımda bulunacak veri ve yargıları işleyen model tabanlı prosedürler bütünüdür [9].

Bilgisayar donanımını üst düzey entelektüel yeteneklerle donatma girişimleri (olasılıksal akıl yürütme yöntemleri, mantık, endüktif sonuç, analoji kanıtı vb.) kullanılan tekniklerin yetersiz olması nedeniyle henüz somut bir sonuç vermemiştir. Entelektüel fonksiyonları uygulayarak kontrol görevlerinin çözümünü otomatikleştirmek için bilinen yöntemler şunlardır [2]:

- Sembolik işleme, algoritmik olmayan problem çözme yöntemleri;
- Sezgisel (Heuristics) yöntemler, deneyimden öğrenilen sezgisel bilgi veya kurallar;
- Sistemin davranışlarını ayarlamasını ve dış ortamdaki değişikliklere tepki vermesini sağlayan makine öğrenmesi;
- Bulanık kümeler;
- Genetik algoritmalar;

- Veri madenciliği;
- Yapay sinir ağlarının ve nörobilgisayarların nesnelerin ve durumların tanınması (sınıflandırılması) ve genelleştirilmesi düzeyinde kullanılması;

Bilişim teknolojisinin son yıllarda hızlı gelişimi ve yaşamın tüm alanlarına girmesinin, bugün nerdeyse tüm işletmelerin Bilgi Teknolojileri (BT) altyapısına sahip olmasını sağladığını söyleyebiliriz.

Modern mikroelektronik gelişim düzeyi, sinir ağı yöntemlerine gerekli donanım desteğini sağlamıştır. Bazı araştırmacılar, KDS'nin sinir ağı platformunda tam olarak uygulanabileceğini savunmaktadır. Yapay sinir ağları (YSA), genellikle tahmin, sınıflandırma, veri ilişkilendirme, veri yorumlama ve veri filtreleme işlemlerinde kullanılmasının yanı sıra, destek ve karar alma sistemlerinde veri girdisi tanıma ve oluşturma, bilginin çıkarımı ve korunması, veri değerlendirme ve karar verme sorunlarının koordineli çözümünü sağlayabilmektedir [5]. KDS sistemlerinin değişen koşullara adapte edilmesi ve görevleri/problemleri çözmesi için büyük ölçekli sinir ağları oluşturmak gerekir. YSA'ya dayalı karar destek sistemleri oluşturmanın temel yönleri göz önüne alındığında, terminolojinin ve mimarinin belirlenmesi önemlidir. Karar verme sisteminin sadece nöronları (nesneleri) değil, tüm sistemi de kontrol ettiği söylenebilir. Karar verme sürecinin özelliği, kontrol sistemi tarafından kendisine sunulan kontrol nesnesinin özellikleri hakkında deneysel bilgi birikimine ve biriken bilgi temeline dayanarak karar vermenin otomatik olarak gerçekleştirilmesidir. Biriken bilginin hacmi arttıkça, yönetimin kalitesi artar. Karar destek sisteminin, gelen bilgilere cevap vermesi değil, mevcut ortamdaki sistemin kendisini geliştirmek için sürekli olarak durumları değerlendirmesi gereklidir.

Havayolu taşımacılığı en hızlı ulaşım şeklidir. Hız ve güvenliğe bağlı istek ve gereksinimler, havacılık teknolojilerindeki gelişmeler, havayolu taşımacılığı sektöründeki şirket ve uçak sayısının artması, rekabetin artmasına neden olmaktadır. Havayolu şirketleri bu rekabete üstün gelebilmek için yatırımları arttırmaktadırlar. Uçak, uçak parçaları veya uçak bakım hizmetlerini satın alırken veya kiralarken, havayolu şirketlerinin en büyük maliyeti filosunun bakımıdır. Bu

sebeple, uçak veya uçağın belirli bir parçasının bakımını mümkün olan en kısa sürede ve en az maliyette yapılması hedeflenmektedir.

Havayolu taşımacılığındaki büyüme diğer ulaştırma yollarına göre daha hızlı olmaktadır. TAV Havalimanlarının Türkiye'de 2020 yılına kadar hava alanlarındaki kalkış ve iniş dâhil uçak operasyonlarında %11'lik bir artış olması beklenmektedir. 2018 yılında yolcu sayısı %31 artarak 152 milyona ulaşmıştır. %24 artış gösteren ticari uçuş sayısı 2018 yılında 1024 bin olarak gerçekleşmiştir. Dünyadaki en iyi 10 havayolu şirketi arasında yer alan Türk Hava Yolları 2020'de uçak sayılarını 3 katına çıkartacaklarını belirtmektedir [20]. Havayolunun emniyeti, havacılık otoritesinin uyguladığı denetime bağlıdır. Her ülkenin havacılık otoritesi farklıdır. Havayolları, hem kendi ülkelerinde, hem uçtukları ülkelerin havacılık otoriteleri tarafından denetlenir. Türkiye'de Sivil Havacılık Genel Müdürlüğü (SHGM) uçakları denetleyerek sorun bulunmadığı takdirde uçuşa elverişlilik sertifikası vermektedir. Yaşlanan uçağın uçuşa elverişliliğini sağlamak ve onarılmış yapıların kalan ömrünün doğru tahmin edilmesi için iyileştirilmiş yöntemler ve yeni sistemler geliştirilmesine ihtiyaç vardır. YSA gibi yeni tahmin yöntemlerinin motorlar, montajlar, parçalar ve bileşenler için sökme ve inceleme tarihlerinin tahmininde faydalı olabilirliği incelecektir.

Havayolları sürekli olarak düzenli işleyişin bozulması ile karşı karşıya kalmaktadır. Hava muhalefeti, hava trafik yönetimi kararları, uçuş güvenliği veya uçak mekanik arızaları gibi olaylar genellikle havayolunun programında gecikmelere veya uçuş iptaline neden olur. Bu tür aksaklıklar havayollarını maddi ve manevi zarara uğratmaktadır. Bu aksaklıkların havayolu programı üzerindeki etkisinin azaltılması, maliyetleri önemli ölçüde azaltacaktır. Benzer aksiliklerden en az zararla kurtulmak için KDS'ler ve akıllı sistemler geliştirilmektedir. Havacılıkta bu tür programların rolü çok önemlidir, hatayı minimuma indirgeyip en uygun çözümü sunmaları gerekmektedir.

Günümüzde havayolu işletmeciliğinde bilginin tüm bölümlere eksiksiz ve zamanında aktarılması büyük önem arz etmektedir. Özel sistemler sayesinde havayolları kendi verilerinin toplanmasını, hesaplanmasını, yönetilmesini, işlenmesini, sorgulanmasını ve analiz edilmesini sağlar. Havayolunun üretim ve yönetim faaliyetlerinin bir

parçası olarak, kurumun faaliyetlerinin göstergelerini elde etmek için verilerin işlenmesi ve analiz edilmesi havayolunun operasyonel faaliyetleri hakkında biriken istatistiklerin tek bir bilgi sisteminde toplanmasını gerektirir.

Havacılık sektörü için geliştirilen KDS optimizasyon sistemleri hali hazırda kullanılmaktadır. Örneğin, Bratu ve Barnhart 2006 yılında, hangi uçuş bacağının erteleneceğine ve hangisinin iptal edeceğine karar vererek uçak, mürettebat ve yolcu için eşzamanlı olarak kurtarma planları geliştiren havayolu tarifesi kurtarma modelleri ve algoritmaları önermektedirler. Amaç, müşterek havayolu işletme maliyetlerini ve tahmini yolcu gecikme ve aksama maliyetlerini en aza indirmektir. Maliyetleri dengelemek, potansiyel olarak müşteriyi elde tutmak, sadakatini artırmak bu şekilde havayolu karlılığını arttırmak hedefler arasındadır [20]. Abdelghany 2008'de, havayollarına tüm uçuş kaynaklarını uyarlayan proaktif bir kurtarma planı geliştirme imkânı sunan bir KDS önermektedir. Sistemin benzetim modeli, bozulan operasyonların ciddiyetini dikkate alarak bozulan uçuşları listelerken, optimizasyon modeli yeni uçuş planı ve olası kaynak değişikliklerini araştırarak uçuş gecikmelerini ve iptallerini en aza indiren etkin geri kazanım planları sunmaktadır [10].

Tekiner 2009'da, düzenli uçuş programına yeni uçuşların eklenmesi durumundaki uçuş dizisi bulma problemini incelemiştir. Çalışmada, maliyet artışları kabul edilebilir seviyelerde tutularak, önerdikleri model ile elde edilen sonuçlar geleneksel yaklaşımla karşılaştırılmıştır [11]. Armacost 2002'de, hızlı paket servisi için en iyi veya en iyiye yakın uçuş ağını tasarlamak için model ve algoritmalar hazırlamıştır. Kargo işletmesinde yapılan çalışmalar sonucunda operasyon maliyetlerinin %7 ve gerekli filo boyutunun %10 oranında azaltıldığı rapor edilmiştir [12]. Daskin ve Panayotopoulos 1989'da, bir dizi karar sürecinden oluşan uçak çizelgeme problemini incelemişlerdir. Tek bir topla-dağıt (hub and spoke) ağ yapısında uçakları rotalarına atayarak işletme karının çok büyüyebileceğini göstermiştir. İlk olarak, filoların uçuş rotalarına atanması gerçekleştirilir, ardından her uçak için bakım kısıtları dikkate alınarak uçuş rota planları oluşturulur. Problem, karma tamsayı doğrusal programlama problemi olarak modellenmiş ve Lagrangian gevşetmesi ile çözülmüştür [13]. Feo ve Bard

1989'da, hem bakım merkezlerinin yerleştirilebileceği hem de A tipi kontrol gereksiniminin daha iyi karşılanabileceği uçuş programı geliştiren model sunmuştur. İki aşamalı sezgisel bir yöntem kullanılmıştır [14]. Kabbani ve Patty 1992'de, uçak rotalama problemini küme belirleme problemi olarak formüle etmiştir. Çalışmada bakımların her üç günde bir gerçekleştirildiği kabul edilmiştir [15]. Hane 1995'de, filo atama problemini çözmek için bir model önermiştir. Problemi çözmek için kullanılan yöntem, iç nokta algoritmasıdır [16]. Barnhart 1998'de, filo atama ve uçak rotalama problemini dal-bedel (branch-and-price) yaklaşımını kullanarak çözmüştür [17]. Gopalan ve Talluri 1998'de, uçak rotalama problemini US Airways için modellemiştir. Üç günlük bakım kontrolü ve dengeli kontrol gereksinimi için polinom zamanlı bir algoritma kullanmıştır [18]. Cordeau 2001'de, uçak rotalama ve ekip çizelgeleme problemine çözüm için Bender ayrıştırması yaklaşımını kullanmıştır [19]. Sriram ve Haghani 2003'de, bakım çizelgelemesi ve uçağın yeniden atanması problemini incelemiştir. Yazarlar, modellemede hem A hem de B tipi kontrolleri dikkate almış ve çözüm için rassal araştırma ve derinlik önce arama (depth first search) yaklaşımlarını karma olarak kullanan bir sezgisel yöntem kullanmıştır [21]. Sarac 2006'da, bakım aralıkları (slot) ve bakım istasyonlarında kullanılabilir adam-saatini dikkate alan operasyonel uçak bakım rotalama problemini modelleyerek çözmüştür [22]. Orhan 2007'de, uçak rotalarının bakım gereksinimleri ile birlikte ele almış, böylece uçakların etkin kullanımının yanı sıra bakım maliyetlerinin en aza indirme hedeflemiştir. Belirtilen hedef doğrultusunda, uçakların bakıma girmeden önceki yasal kullanılabilir uçuş zamanlarını da en aza indiren çok-amaçlı karma tamsayı doğrusal bir modelleme yaklaşımı önermiştir. Yaklaşım, uçuş sayılarını veya sürelerini de karar vericinin öncelikleri doğrultusunda dengeleyerek uçak kullanım oranları ve bakım maliyetlerinin olabildiğince eşit tutulmasını sağlamıştır [23]. Ryan 1992'de, uçuş ekibinin memnuniyetini en üst seviyeye getirmek için ekip atama problemini, önce ilkil (primal) simpleks algoritmasıyla, sonra tam sayı sonuç elde etmek için dal ve sınır yaklaşımıyla çözmüştür [24]. Özdemir 2009'da, literatürdeki uçuş dizisi bulma problemi üzerine yapılan çalışmaları inceleyerek, hibrid bir metot geliştirmiştir [25]. Arıkan 2017'de, konik bir kuadratik karışık tamsayı programlama formülasyonu kullanarak tamamen entegre havayolu

kurtarma problemini çözen bir optimizasyon sistemi önermektedir [26]. Tammineni 2007'de, gaz türbinli uçak motorunun performans parametrelerinin, ön tasarım aşamasında sistem/alt sistem konfigürasyonunu ve ilgili maliyetleri etkilemedeki rolü üzerinde çalışma yapmıştır. Kullanıcı tarafından performans parametrelerinin özelliklerine dayanarak bir sistemin/alt sistemin konfigürasyonları için maliyet tahminlerini sağlamak üzere bir hibrit karar ağacı çıkarım mekanizması kullanan bir uzman KDS sağlanmıştır [27].

Havayolu sektöründe işletmeler bilgisayar destekli çözüm sistemleri ve yöntemleri kullanmaktadırlar. Bu sistemler ve yöntemler işletmelere ciddi rekabet üstünlüğü sağlamaktadır. İşletmelerin KDS kullanarak maliyetlerinde önemli kazanımlar sağladığını ortaya koymaktadır.

Genel olarak, bu sistemler havayolu işletme maliyetlerini ve yolcu maliyetlerini en aza indirmede matematiksel bir kabiliyete sunmaktadır.

Havacılık sektörü, diğer sektörlere nazaran çok daha fazla çizelgeleme ve planlama süreci barındırmaktadır. Asıl amaç, kaynak planlamasını yapabilmek**,** kar artırmak ve problemleri en aza indirgemektir. Havayolu işletmeleri kendi problemlerin çözümü için en kısa sürede çözüm sağlayabilecek sistem arayışında olduklarından, gelişmiş teknolojilerle entegre olan karar yöntemlerinin kullanılması şirketlere büyük kar olanağı sunar. Çözüm sürelerini iyileştirecek ve en aza indirecek yeni sistem ve yöntemlerin geliştirilmesi, karşılaşılan problemlerin uygun şekilde ve daha kısa sürede çözülmesini sağlamış, şirketlerin maliyetlerini düşürerek ciddi oranda kar ve tasarruf olanağına neden olmuştur.

Halen Federal Aviation Administration (FAA) tarafından geliştirilmekte olan Emniyet Performans Analiz Sistemi (SPAS), FAA denetim faaliyetlerini destekleyen, müfettişlerin olası sorunlu alanları değerlendirmesi için analitik bir araç sunmaktadır [28]. SPAS, denetçilerin mevcut FAA bakım veri tabanlarına erişmesini sağlayarak yüksek güvenlik riski taşıyan ve izlenmesi gereken havayolları tespit eden bir karar destek sistemi olarak işlev görecektir.

Literatür taraması yoluyla gerçekleştirilen derleme niteliğinde KDS, problem tanımaya, problem yapısına, bilgilere, istatistiksel araçlara ve metodolojik bilgilere

daha kolay erişim sağlayarak bireysel öğrenmeyi geliştirmeyi amaçlamaktadır. KDS, daha kolay ve daha hızlı alternatiflerin oluşturulmasını ve mevcut karar alma sürecinde eksikliklerin farkına varılmasını sağlamak için tasarlanmıştır [29]. İyi bir KDS, problemi çözmek için karar alma alanını inceler, sonuç çıkartır, açıklamalar sağlar ve sonuçlara nasıl ulaştığına dair tüm bilgileri saklar.

YSA, verileri izleme ve potansiyel sorunları işaret edebilecek eğilimleri tespit etmede yararlıdır, ayrıca; modelleme, sınıflandırma ve tahmin gibi yeteneğine sahiptir. YSA KDS'de kriterler ve alternatifler arasındaki sıralamanın veya seçimin belirlenmesinde kullanılmaktadır. YSA özellikle dış uyaranları kavrama ve neden-sonuç ilişkilerini değerlendirmeye çalışmanın problem tanıma ve tanı karar rutinlerini desteklemek için uygundur. Bu tez kapsamında da motor bakım sürecinin takibi ve planlanması için yapay sinir ağlarını kullanan yöntem önerilmiş ve uygulanabilirliği gösterilmiştir.

Sistem, karşılaşılan probleme karşı alınacak aksiyonları, tedbirleri ve olasılıkları tespit ederek uzman kişinin kararına yardımcı olacaktır. YSA'nın görevi, önceden yapılmış değerlendirmeler sonucu ortaya çıkan durumları ve problemleri analiz ederek karşısına çıkan yeni durumlara ve problemlere karşı çözüm üretmektir. Çalışmada geliştirilen teknik yeni alınan kararları veri tabanında saklayarak daha kaliteli ve güvenli kararlar alınmasını amaçlamaktadır.

Havacılık alanında bulanık mantığın pek çok uygulaması bulunmaktadır. Uçuş durumu ve uçuş parametre verileri arasındaki ilişkiye dayanarak pilota uçuşla ilgili ilave durum bilgisi iletiminde [30], uçuş rotalarında parametrelerin hesaplanmasında, uçağın uçuş kontrol sisteminde ve uçuş harita planlarının oluşturulmasında bulanık mantık yöntemi kullanılmaktadır [31]. Kelly ve Painter, pilotlara uçuşla ilgili durum bilgisini gösteren ve değerlendiren bir sistem geliştirmişleridir. Uçağın uçuş yönetimindeki muhtemel bozukluklar için algılayıcıların verilerini ve uçakların durum bilgisi izlenerek bir bilgisayarda değerlendirilmektedir. Kelly ve Painter'in geliştirdiği sistemde uçuş durumu ve değişkenler arasındaki ilişkiler bulanık mantık kullanılarak değerlendirilmektedir [32]. Uçuş kontrolünde sıkça karşılaşılan sensör kaynaklı gürültüyü istikrarlı hale getirmek için bulanık mantık kullanılmıştır [33]. Boskoski ve arkadaşları

tarafından tasarlanan bulanık denetleyici, uçağın otomatik iniş kontrol sisteminde kullanılmıştır [34]. Meyer ve arkadaşları, havayolu taşımacılığında büyüyen trafiğe çözüm olarak güvenli bir yönetim sağlayabilmek için bulanık mantığa dayalı sistemlerin tercih edilmesi önerilmektedir [35]. Özdemir, uçak sisteminin yuvarlanma açısını kontrol etmek için bulanık mantık kullanmıştır [36]. Vijayan, John ve arkadaşları tarafından dinamik uçak parametre değerleri ve çevresel faktörler dikkate alınarak uçuş harita planlaması yaklaşımı önerilmiştir. Bu yaklaşımla yapılan uçuş harita planlamalarının fayda sağladığı öngörülmüştür [37].

Yapılan çalışmada; motor bakım planlanma probleminin optimal olarak çözülebilmesi için, yapay zekâ teknikleri kullanarak karar destek sistemlerinin tasarımı ve gerçekleştiriminin nasıl yapılacağı incelenmiş ve genel bir karar destek tasarım modeli önerilmiştir.

1.2 Tezin Amacı

Tezin amacı bulanık mantık ve sinir ağları yöntemlerini kullanarak bir KDS tasarlamak, motorların durumunu değerlendirmek, riskleri en aza indirgemek ve bilinçli bir karar vermektir. Bu amaç doğrultusunda aşağıdaki konular incelenecektir:

- Motorların takım/söküm ile alakalı güncel sorunlarının araştırması.
- Mevcut algoritmaların ve risk değerlendirme yöntemlerinin analiz edilmesi.
- Veri analizlerin yapılması ve sonuçların çıkarılması.
- Kapsamlı bir KDS modelinin geliştirmesi.

Havacılık şirketlerinde iş akışında riskleri izlemek ve azaltmak için daha modern araçlar kullanmak gereklidir. Örnek olarak riskleri hesaplayan, değerlendiren, kontrol eden ve yöneten bilgi sistemleri verilebilir. Asıl sorun uçakların üzerinde takılı olan motorların bakım süreci veya söküm tarihi gelmeden uçak üzerinden sökülerek yerde yatmalarını engellemektir.

Filo yönetiminin işlevleri:

- Uçuş planlamasını takip etmek.
- Uçak bakım planlamalarını yapmak ve kontrol etmek.

- Parça takım/söküm işlemlerini gerçekleştirmek.
- Kiralık uçakların teslim ve iade işlemlerini takip etmek.
- Filo değerlendirmesini yapmak.

Uçak bakım maliyetlerinin doğrudan işletme maliyetlerinin %20'sini oluşturduğu göz önüne alındığında bu maliyetlerin kontrol altına alınarak daha etkili yöntemin sağlanması havayollarının rekabet gücünü olumlu etkileyeceği açıktır. Havayolları Bakım, Onarım ve Yenileme (BOY) şirketlerinden aldıkları servis oranları her geçen gün artmaktadır. BOY faaliyetleri:

- Uçak hat bakımı;
- Uçak ağır bakımı;
- Motor bakımı;
- Parça bakımı;
- Onarım servisleri.

BOY ve havayolu şirketleri, doğru bir şekilde bakım sürecini yönetip takip edebilecekleri sistemlere ihtiyaç duymaktadır.

Havayolu endüstrisi küresel standartlara ve gereksinimlere uygun olarak gelişen ileri bir teknolojiye sahiptir. İş süreçlerinin yapısının karmaşıklığı yüksek sorumluluk gerektirir. Bu sektörde, bilginin aktarılmasındaki en ufak bir başarısızlık çok ciddi sorunlara yol açabilir. Yolcu ve yük trafiğinin hızlı bir şekilde artması, bir yandan uçakların kullanım ve bakımının optimize edilmesine yüksek talepler oluştururken, öte yandan, sivil havacılıkta, havayollarının ve havaalanlarının müşteri odaklı çalışması için gereksinimler artmaktadır. Buna ek olarak, sektör aynı zamanda küresel bir konsolidasyon eğilimindedir ve çok sayıda birleşme ve satın alma süreci geçirmektedir. Son yıllarda havacılık sektörünün tüm yeni ihtiyaç ve talepleri, BT için de yeni gereksinimler yaratmıştır. Havacılık şirketleri BT altyapılarını geliştirerek ve sürekli modernize ederek güvenilirliği sağlar. Buna ek olarak kendi verimliliğini, rekabetçiliğini, işlerini ve genel yönetilebilirliğini artırmanın ve maliyeti düşürmenin yolunu BT'de aramaya başladılar. Operatörlerin taşımacılık hizmetleri aşağı yukarı aynıdır. Yolcuların konfor seviyesini artıran ve çekiciliğini etkileyen yeni hizmetler özellikle BT

üzerine kuruludur. Şirket kaynaklarında yeni yönetimler verimliliği arttırır. Örneğin, özel bir BT sisteminin yardımıyla uygun uçuş planını oluşturarak, yılda 200-300 bin$ kadar havayollarına tasarruf sağlanabilmektedir. Ayrıca piyasanın gelişimi, havayolu şirketlerinin değişikliklerine hızla yanıt vermelerini ve müşterilere anında yeni hizmetler sunmalarını gerektirir. Tüm bunlar esnek, ölçeklenebilir, hataya dayanıklı bir BT altyapısı olmadan mümkün değildir. Yüksek teknolojiler, yolcu trafiğindeki büyümenin temelini oluşturmak ve mevcut durumda hayatta kalmak için muazzam bir potansiyele sahiptir. Büyük havayolları SAP veya Oracle tarafından sunulan ERP sistemlerini tercih etmektedirler (aynı zamanda alt sistemler onlarla başarılı entegrasyon için yeterli esnekliğe sahip olmalıdırlar). ERP sistemleri uçuşun zaman, hız, rakım - operasyonel parametrelerinin onarım ve bakım için belirli bir kaynak ve malzeme konfigürasyonuyla uçağın ihtiyaçlarındaki problemi çözmek için tasarlanmıştır. Havacılık sektöründeki ERP sistemleri; uçuş problemlerini ve uçak arızalarını, uçağın ihtiyaçlarına göre onarım ve bakım işlemlerini takip etmek, kaynak ve malzeme tedarik işlemlerini planlamak için tasarlanmıştır. ERP'nin havacılıktaki temel görevi üç ana süreç entegrasyonundan oluşur. Bunlar: uçuş yönetimi, onarım yönetimi (hangar yönetimi dâhil) ve diğer havayolu hizmetlerinin yönetimidir (çalışanların giriş-çıkış saatleri, eğitimler vb.).

Tablo 1.1 Dünya Havayollarında BT Sistemleri

Şirket	**Ülke**	**Çözüm**
Air France	Fransa	mySAP ERP (finans, muhasebe, bütçeleme, BOY) Sirax AirFinance Platform (Lufthansa Sistemleri) - faturalandırma Vision64 (Metrix Systems) –CSM, BT yönetimi OnAir - gemide mobil iletişim (sms, mms, e-posta)
Air India	Hindistan	Oracle E-Business Suite
Alitalia	İtalya	SAP ERP, CRM Sirax (Lufthansa Sistemleri) - Finans MicroStrategy İş Zekâsı Platformu - Analytics, BI AMOS (İsviçre Havacılık Yazılımı) – BOY

AeroFlot	Rusya	SAP R3 (muhasebe, raporlama, depo) HP Open View (BT izleme), IBM WebSphere, AMOS - BOY
British Airways	İngiltere	SAP ERP MRO-BOY Altéa Customer Management Solution (Amadeus) - rezervasyon sistemi, kayıt sistemi Airfare Insight (SITA) - uçak yolculuğu için piyasa oranlarını izlemek için bir sistem SAN (CA) - BT izleme, veri ambarı CommuniGate Pro (CommuniGate Sistemleri) - uçakta e-posta
Delta Air Lines	ABD	SAP ERP MRO - BOY Gogo Inflight Internet (Aircell) – uçakta Wi-Fi ActiveMatrix (Tibco) – COA
S7	Rusya	Oracle E-Business Suite (muhasebe, finans, lojistik, bakım, depo)
Singapore Airlines	Singapur	SAP ERP
UIA	Ukrayna	AMOS – BOY

Havayolu şirketleri maliyetlerini düşürmek, sektördeki tüm ihtiyaçlarını ve süreçlerini takip etmek için sistem arayışındadırlar.

Aranan özellikler:

- Filo yönetimi (genel durum takibi);
- Birden fazla kaynaktan elde edilen verilerin birleştirilmesi ve dönüştürülmesi;
- Uçuş yönetimi;
- Stok yönetimi;
- Bakım ve onarım yönetimi;
- Proje ihtiyaçları;
- Malzeme optimizasyonu;
- Veri yedekleme;
- Yönetim muhasebesi;
- Kalite gereksinimleri, vb.

Havacılık sektörü için geliştirilen yazılımlarda genellikle ADA, C/C++, Oracle ve Java dilleri tercih edilmiştir [38, 39, 40].

Kurumların bu süreçte desteklenmesini sağlayacak bir KDS ihtiyacı derinden hissedilmektedir. Bu nedenle, bahsedilen özellikleri içeren bir KDS tasarımı hedeflenmiştir. İhtiyaca yönelik olarak, birçok kısıtlamanın ve çoklu amaç fonksiyonunun olduğu karmaşık bir yapıda, kısıtlı kaynaklar altında optimizasyon probleminin çözüldüğü bir karar destek sistemi geliştirilecektir. Bu karar destek sisteminin sağlam çözümler sunması, kısıtlı kaynakları etkin ve verimli kullanarak bakım kapasitesini ve genel hizmet kalitesini iyileştirmek için performans artışı hedeflenmektedir.

1.3 Hipotez

Uçak motorlarının bakım maliyeti yüksektir. Motor bakım süreleri diğer uçak bakım sürelerine nazaran daha uzun zaman alan işlemlerdir. Havacılık alanında, uçak bakımlarının en kısa sürede tamamlanması ve uçağın yeniden operasyona döndürülmesi önem arz eder. Günümüzde havayolu şirketleri ve taşımacılık yapan firmalar, bakım kuruluşlarından planlanan bakımın en kısa sürede tamamlamasını istemektedirler. Bu talebe dayanarak, bakım programı oluşturulan sistemlerine ihtiyaç duyulmaktadır.

Uçak motorlarının bakım süreci, ilgili personelin takibi ile geleceğe yönelik yeterli planlama olmaksızın gerçekleştirilebilmekteyken, operatörler için daha düşük maliyetli olan söküm/takım planı oluşturan ve bakım sürecinin optimum fayda sağlayacak şekilde planlanmasına olanak sağlayacak bir KDS geliştirilebileceği düşünülmektedir. Teknolojik altyapı gerektiren bu sistemlerin maliyetleri azaldıkça işletmeler tarafından daha çok kullanılmaya başlayacağı beklenmektedir.

2
Materyal ve Yöntem

Şirketler için bilginin doğru ve hızlı şekilde analiz edilmesi oldukça önemlidir. KDS işletmelere verilerine anında ulaşabilmenin yanı sıra gerçek zamanlı bilgi üretebilme ve analiz edebilme imkânı sunar.

2.1 KDS Tanıtımı ve İncelenmesi

KDS'yi ilk Amerikalı bilgisayar bilimcileri geliştirmeye başladı (T. Gruber, N. Guarino, M. King, M. Ushold, Y.Zorgios, G. Rzevski) [79].

Genel olarak KDS için farklı tanımlar yapılsa da hepsi aynı işlevsellik altında yer alır. KDS, veri madenciliği, modelleme ve görselleştirme araç setini kullanarak karar vericilere kararların alınmasında yardımcı olur. Karar verme surecini hızlandıran ve kolaylaştıran, etkileşimli ve ayarlarda esnek, kalite açısından istikrarlı, modelleme imkânına sahip, esnek bir kullanıcı arayüzü (GUI) barındıran bir sistemdir.

KDS, insanların karar almasına yardımcı olan ve işin nasıl yapılacağının bilinmediği sahalarda çalışabilmeyi sağlayan sistemdir. KDS, planlanmış ya da planlanmamış durumlarda karar vermeyi destekleyen ve elimizdeki veriyi, bilgi ve örnek yaklaşımlar sayesinde beceriyle kullanabilmemizi sağlar.

KDS, kullanıcıların daha iyi kararlar alabilmeleri için verilere ve/veya modellere erişmelerini sağlar.

Başarılı bir KDS tasarlamak ve hayata geçirmek için en önemli kriter gereksinimleri belirlemektir. Başka bir deyişle, iş hedefleri ve ihtiyaçlar doğrultusunda analizi yapabilmek için gereken bilgilerin operasyonel sistemde güncel olarak beslenmesi gerekir.

2.1.1 KDS Mimarisi

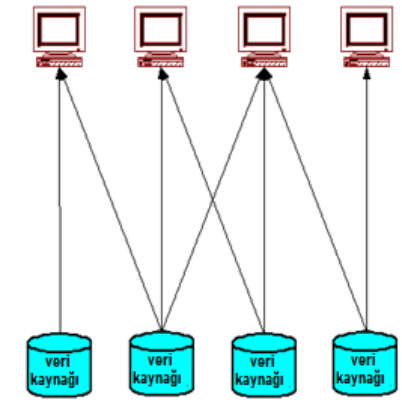
Dört tür KDS mimarisi vardır [3]:

- Fonksiyonel KDS.

- Bağımsız veri depolama.
- İki katmanlı veri depolama.
- Üç katmanlı veri depolama.

Fonksiyonel KDS

Fonksiyonel KDS, mimari açıdan en basit çözümdür. Bu sistemler pratikte özellikle, bilgi altyapısı ve analitik seviyesi yeterince gelişmemiş kuruluşlarda daha yaygındır.



Şekil 2.1 Fonksiyonel KDS

Fonksiyonel KDS, analizin operasyonel sistemlerden elde edilen verileri kullanılarak yapılması ile ortaya çıkar.

Avantajları:

- Veri ambarlarına göre oluşturma kolaylığı.
- Düşük maliyetlidir.
- Veri ambarına göre daha güvenli ve esneklik.

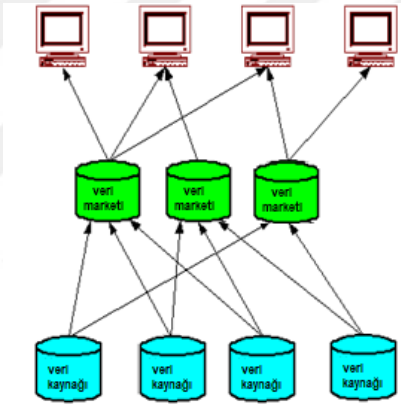
Dezavantajları:

- Tek veri kaynağı, sistemin cevaplayabileceği soru çeşitliliğini daraltır.
- Stratejik karar vermedeki rolleri bakımından bu operasyonel sistemlerin kalitesi düşüktür. Veri temizleme aşamasının olmaması nedeniyle genellikle düşük kalitededir.
- İşletim sisteminin yavaşlamasına ve sistem arızasına sebep olur.

Bağımsız Veri Marketlerini (Data Mart) Kullanan KDS

Veri ambarlarının daha küçültülmüş ve hedefe yöneltilmiş alt kümelerine veri marketleri (Data Mart) denilir [82]. Veri marketleri, veri ambarlarının alan olarak daha daraltılmış halidir. Veri ambarları gibi bir problemin bütününe değil belirli bir kısmına yönelik bir bakış sağlar. Genelde tek bir grup veya bölümün verilerini işleyerek ana kümeden bağımsız çalışabilir.

Veri ambarları, bir şirketin tüm verilerini barındırırken veri marketleri daha özelleşmiş bir yapıdadır. Bu bütünleşik ve biraz daha karmaşık yapıdan ziyade, sadece belirli bir iş ile ilgili olan kısmın verilerini içerir. Örneğin, bir şirketteki departman sadece kendine ait olan verilere ulaşmak ve bunun üzerinden analiz yapmak isteyebilir. İlgili departman için veri marketleri hazırlanır ve böylece bütün yapıdaki karmaşıklıkla uğraşmasına gerek kalmaz.



Şekil 2.2 Bağımsız Veri Marketleri

Avantajları:

- Hızlı uyarlanabilir.
- Belirli sayıda soruya cevap verecek şekilde tasarlanma.
- Hızlı performans sağlar.

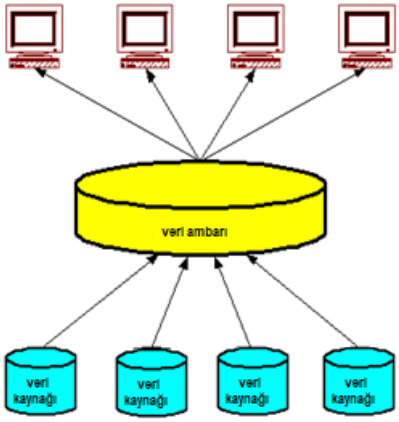
Dezavantajları:

- Farklı veri marketlerinde veri kaydı özelliği, verilerin çift olmasına yol açar. Bu da verilerin saklanması için ek maliyet gereksinimine neden olur ve veri tutarlılığını sağlamak zorlaşır.

- Çok fazla veri kaynağı varsa, veri marketini doldurma sorunu oluşur.
- Veri bütünlüğü yoktur.

İki Katmanlı Veri Ambarına Dayanan KDS

Veri depolama işlemi iki ana kavrama dayanır. Tüm verileri tek bir veri tabanında tutar ve veri gruplarının uygulamalarının dağılımını sağlar. İki katmanlı veri ambarı mimarisi, merkezi bir veri ambarı oluşturmadan veri marketleri oluşturma fikrine dayanır. Veri marketleri oluşturulurken, veri ambarı mimarisinde kullanılan kurallar uygulanır. İki katmanlı veri ambarı, şirket içinde bilgi aktarımını sağlamak için oluşturulur.



Şekil 2.3 İki Katmanlı Veri Ambarı

İki katmanlı veri ambarı, kurumun/şirketin tüm veri tanımlarının ve süreçlerinin üzerinde hemfikir olması gerektiğini gösterir.

Avantajları:

- Veri tek nüsha olarak saklanır.
- Depolama maliyetleri en aza iner.
- Birden çok veri kopyasının uyumlanma sorunu yoktur.

Dezavantajları:

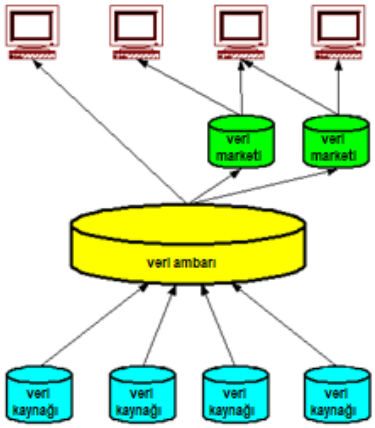
- Veriler ihtiyaçlarını destekleyecek şekilde yapılandırılmadığından, bireysel kullanıcıların veya kullanıcı gruplar için kullanım zorluğu vardır.
- Sistem performansı ile ilgili olası sorunlar vardır.

– Kullanıcıların verilere erişim haklarını sınırlandırmada ortaya çıkan olası zorluklar vardır.

Üç Katmanlı Veri Ambarına Dayanan KDS

Üç katmanlı mimari, genellikle tam donanımlı bir kurumsal veri ambarı oluşturmak için kullanılır.

En alt katmanda veri kaynakları yer alır. İkinci katman veri tabanıdır. Burada ilk katmandan toplanan veriler saklanır. Üçüncü katman ise veri marketlerinden oluşur. Buraya veriler ana veri tabanından (veri kaynağı) gelir ve kullanıcılar genelde veri marketleri ile çalışır.



Şekil 2.4 Üç Katmanlı Veri Ambarı

Veri ambarı işlenmiş ve özelleştirilmiş verileri tutar. Veri marketleri veri ambarların alt kümeleridir. Şirketin bireysel bölümlerinin sorunlarını çözmek için tasarlanmıştır. Veri marketlerinde yeterli veri bulunmadığı takdirde veya kullanıcılar tüm veriyi elde etmek istediklerinde veri ambarına ulaşabilirler.

Avantajları:

- İyi yapılandırılmış, tanımlanmıştır.
- Verileri tek bir veri kaynağından sağlanır.
- Veri marketlerini yaratmak daha basittir.
- Veri marketleri veri ambarı ile senkronize edilir.
- Yüksek performans sağlar.

Dezavantajları:

- Veri fazlalığı vardır.
- Depolama alanı büyümeye başladıkça, yönetmek zorlaşır ve maliyeti artar.
- Yapılandırmak zor ve uzun sürer.

KDS mimarisinin açıklanmasına ilişkin birkaç yaklaşım vardır. KDS 4 ana katmana bölünebilir [9, 10]:

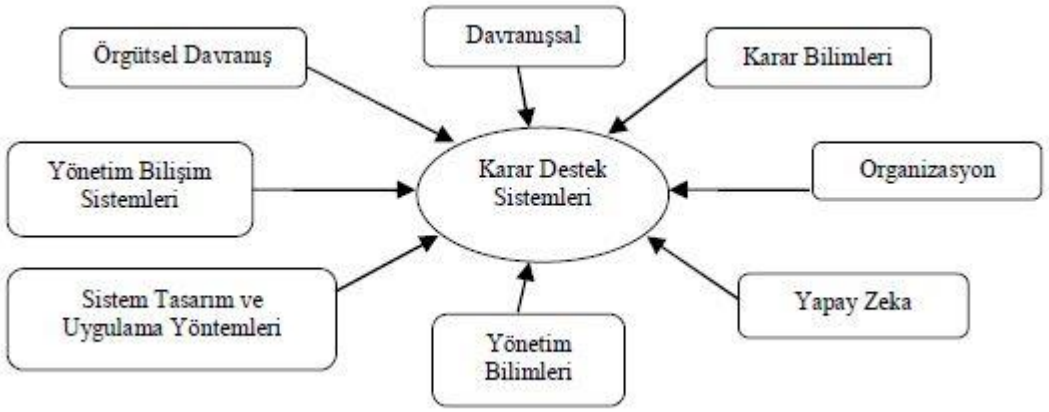
Tablo 2.1 KDS Mimarisi

Katman	**İşlevsellik**	**Araçlar**
Arayüzey (Interface)	Etkileşimlilik; Görselleştirme;	Mobile App,Web portal, Desktop App, Flask, QlikView,Power BI, Tableau
Modelleme (Modeling)	İstatistiksel modeller ve makine öğrenmesi; Sayısal modeller; Oyun teorisine dayanan modeller;	Python,Jupyter,SPSS,Apache Spark,MATLAB,SAS, Statistica
Veri Madenciliği (Data mining)	Veri akışı sağlama; Veri işleme; Uzman Değerlendirme;	Apache Spark,Mongo DB, MySQL, SAS,Oracle,Alteryx, SAP
Veri Toplama (Data collection)	Web-Crawling; Sensors; API.	Python,IBM Watson,Azure Data Lake, Amazon webservices S3,Hadoop

KDS oluştururken aşağıdaki adımlar izlenilir:

1. Etki alanı analizi (aslında, KDS'imizi kullanacağımız yer).
2. Veri toplama.
3. Veri analizi.
4. Model seçimi.
5. Uzman analizi / modellerin yorumlanması.
6. Modellerin tanıtımı.
7. KDS'nin değerlendirilmesi.
8. KDS kurulumu/uygulanması.
9. Geri bildirim toplama (herhangi bir aşamada).

Karar Destek Sistemlerinin ilgili olduğu bilim dalları Şekil 2.5'de gösterilmiştir.



Şekil 2.5 KDS'in İlgili Olduğu Bilim Dalları [80]

KDS iki şekilde değerlendirilebilir; ilk, yukarıda incelediğimiz özelliklere göre, ikincisi, kriter kontrol listesine göre değerlendirmektir. Örneğin:

Tablo 2.2 KDS Değerlendirmesi

Sistem	Tepki süresi, Veri ile çalışma, Çıktı kalitesi, İnsan-Makine arayüz kalitesi, Kaynak verimliliği
Görev	Karar verme için harcanan zaman, analizde alternatiflerin bütünlüğü, birinci ve ikinci türden hatalar
Gelişim	Esneklik, ölçeklenebilirlik, koşullar değişikliklerinde hızlı değişiklik olasılığı
Faaliyet/İş	Geliştirme maliyeti, parasal etki(business case), parasal olmayan etki, alternatif maliyeti(KDS kullanılmazsa)

Bu basit bir kontrol listesi (check-list) örneğidir. Her şirket kendi ihtiyaç ve gereksinimlerine göre kendi kontrol listesini oluşturur. Liste ihtiyaca ve probleme göre değişiklik gösterebilir.

Makine öğrenmesi ve oyun teorisi vb. algoritmalar nerde yer alır diye sorgulanırsa, her yerde diyebiliriz. Ama en önemli yeri modelleme ile ilişkili katman içerisindedir. KDS'yi tasarlarken kullanacağınız algoritmanın seçimi önemlidir. Basit bir KDS için kapsamlı ve zor bir yöntemler yerine klasik yöntemleri seçmemizde yarar vardır (bu durum sistemi boş yere karmaşık ve zor hale getirir).

Stratejik kararlar almak için oluşturulan KDS'de oyun teorisini kullanmak daha uygundur. Genel olarak RandomForest, CatBoost ve Stackelberg denge modeli kullanılır.

Avantajları:

- Geçmiş verilere gerek yoktur.
- Model denge hesabı vardır.
- Sonuç hesaplanır.

Makine öğrenmesi, oyun teorisine karşılaştırıldığında

- Geçmiş verilere ihtiyaç duyar.
- Veri sentezi yapılır.
- Tahmin yürütülür.

IBM'in Tivoli KDS'sinde bağlantı noktası numara paylaşma algoritması kullanılmıştır. Bu sistem süper bilgisayarlarınızın durumunu belirlemenize olanak sağlar (Watson ilk sırada): Watson performansıyla ilgili bilgiler, günlükler incelenerek elde edilir. Kaynak kullanılabilirliği, bakım ihtiyaçları vb. tahminler elde edilir [61].

Finlandiya Ulaştırma Bakanlığı, Fin kökenli sensör üreticisi VAISALA (hava ölçümü ve tahmini yapan çeşitli endüstriler için çeşitli ortam ölçüm cihazları üreten, havacılıktan, su bilimine kadar çeşitli servisleri sağlayan, Fin kökenli şirkettir), KDS sistemini geliştirerek hangi dönemlerde buzlanma olacağını tahmin ederek buzlanmayı önleyerek kaza riskini en aza düşürmeyi hedefler [62].

Fin şirketi FOREDATA, bir pozisyon için devam eden seçim aşamasında bir adayın uygunluğu hakkında insan kaynaklarına karar verme sürecinde KDS'yi yardımcı olması için sunar [63].

Dubai havaalanı, hava olaylarını tahmin etmek ve aynı zamanda şehrin sahil şeridini korumak için en son hava durumu teknolojisinin entegrasyonuna dayalı bir KDS geliştirdi. Ayrıca, havaalanında kargo terminalinde, şüpheli yükü/valizi belirleyen bir KDS tasarladı. Yolcunun bilgileri ve gümrük memurları tarafından girilen verilere dayanan algoritmalar, şüpheli yükleri bulur; algoritma yükün hangi ülkeden geldiğini, ambalaj bilgilerini, bildirim alanlarında özel bilgileri vb. değişkenlere dayanarak sonuç üretir [64].

Benzer algoritmalar ABD Dışişleri Bakanlığında da kullanılır. Vize başvuru formu ve diğer belgeler doğrudan elçilikte/konsoloslukta analiz edilir, daha sonra veri görevlisinin ekranına 3 ifadeden biri gösterilir: yeşil (vize ver), sarı (sorular var), kırmızı (vize verme). Amerika Birleşik Devletleri'ne daha önce vize aldıysanız, konsolosluk görevlisinin size verdiği karar tam olarak algoritmanın kurallarla birlikte çalışmasının sonucudur, görevlinin kişisel kararı değildir.

KDS örnekleri çoktur, günümüzde her alanda yer alır; devlet kurumları, eğitim kurumları, telekomünikasyon, sanayi, hastaneler, bankalar, vb.

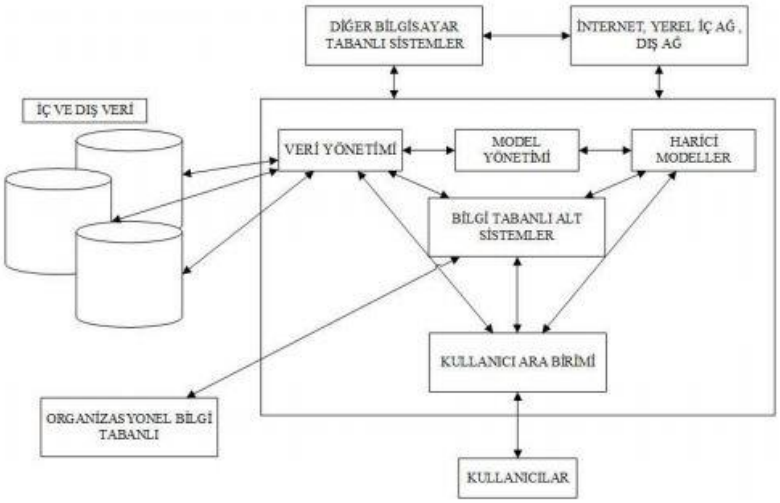
KDS'nın geleceği kararların esnekliğidir. İyi bir KDS tüm yöntemleri/araçları birleştirmeli, farklı işler için farklı modüller kullanılmalı ve farklı kullanıcılar için farklı çıktı arayüzlerine/arabirimlerine sahip olmalıdır. Genel olarak, sinir ağlarının belirsizlik altında karar vermede en uygun yöntem olduğu belirtilmelidir.

2.2 KDS Alt Sistemlerin Tanıtımı

KDS dört alt sistemden oluşmaktadır [9]:

- Veri yönetimi alt sistemi;
- Model yönetimi alt sistemi;
- Kullanıcı ara yüzü alt sistemi;
- Bilgi yönetim alt sistemi [9].

Şekil-2.6'de KDS ana elemanları ve ilişkileri gösterilmiştir.



Şekil 2.6 KDS Alt Sistemleri [81]

1. Veri Yönetimi Alt Sistemi;

Sistemde kullanılan verinin saklandığı ve işlendiği alt sistemdir. Alt sistemi veri tabanı ve veri tabanı yönetim sistemlerini içerir.

- Veri tabanı, kuruluşun uygulama programlarının kullandığı operasyonel verilerin bütünüdür. İşletme tarafından karar verme süreçlerinde kullanılabilecek verinin tutulduğu alt sistemdir. Veriler hem sistem hem kullanıcı tarafından ulaşılabilir olmalıdır. Genellikle Web sunucu üzerinden erişim sağlanır.
- Veri tabanı yönetim sistemi, verilerin fiziksel hafızadaki durumlarını ve kullanıcıların verilere erişimlerini düzenleyen sistemlerdir. Kullanıcılara veriyi oluşturmayı, sorgulamayı, güncellemeyi ve yönetmeyi sağlayan alt sistemdir.

2. Model Yönetim Alt Sistemi;

Sistemdeki modellerin oluşturulduğu, güncellendiği, saklandığı ve sorgulandığı bir alt sistemdir.

3. Kullanıcı Ara Yüzü Alt Sistemi;

Kullanıcı ile sistem arasındaki ilişkiyi sağlayan alt sistemdir. Veri yönetimi ve model yönetimi sistemlerine doğrudan erişim imkânı vardır. Kullanıcılara veri tabanı ve model tabanına erişim olanağı sunar.

4. Bilgi Yönetim Alt Sistemi

Bilgi yönetim alt sistemi bağımsız bileşen olarak diğer alt birimleri destekler. Veri tabanında ve model tabanındaki verileri kullanarak karar alma sürecin ve işlemlerin planlandığı alt sistemdir. Problem çözme süreçleri bu alt sistemde gerçekleşir [9].

2.3 KDS Özellikleri

Genel olarak KDS özellikleri aşağıda sıralanmıştır [51] **:**

- KDS hem verileri hem de modelleri kullanır.
- KDS, yöneticilerin yarı/zayıf yapılandırılmış ve yapılandırılmamış işler için karar vermesine yardımcı olmak için tasarlanmıştır.
- Yöneticilerin kararlarını destekler, değiştirmez.
- Kolay erişilebilirlik sağlar.

- KDS'nin amacı kararların etkinliğini arttırmaktır.
- Geleceği planlamaya yöneliktir.
- Karar verme sürecinin tüm aşamalarını destekler.
- Sorgulama olanağı sağlar.
- Tüm karar türlerine destek sağlar.
- Bağımsız ve ardışık kararlar alır.
- Kullanıcıya karar verme sürecinin tüm aşamalarını kontrol etme olanağı sağlar.
- Farklı düzeylere destek sağlar.
- Değişen şartlara ve karar durumlarına uyum sağlayabilecek esnekliktedir.
- Bağımsız veya bağımlı kararlar için destek sağlar.
- Veri tabanına erişim sağlar.
- Veri inceleme ve çözüm üretmede analitik modeller kullanır.
- Hem bireysel hem grup tabanlı karar verme desteği sağlar.
- Problem çözümünde denetim olanağı sağlar.
- Düzeyler arası karar verme desteği sağlar.
- Kullanım kolaylığı sağlar.
- Veri alışverişi için internet ve ulusal ağlarını kullanır.

KDS'ye neden ihtiyaç duyulur:

- Karar vermenin zorluk ve karmaşıklığı durumunda.
- Çeşitli alternatiflerin doğru bir şekilde değerlendirilmesinde.
- Tahmini işlevsellik ihtiyacında (öngörü ihtiyacı).
- Problem doğrultusunda analizi yapabilmek için gereken bilgilerin sistemde güncel olarak beslenmesinde (girdi veri setleri).

İlk KDS, 60'ların ortalarında - 70'lerin başında Hareket İşlem Sistemleri (Transaction processing system)'den çıkmıştır. O zaman bu sistemlerin fazla bir etkileşimi yoktu. İlişkisel veri tabanı yönetim sistemi (IVYS) üzerindeki sayısal modellemede çok az işlevselliğe sahip olan bir üst yapıyı temsil ederdi. İlk KDS sistemlerine, MIT'de geliştirilen ve geçmiş işlemlere dayalı herhangi bir işlemi simülize eden bir sistem olan DYNAMO örnek verilebilir. IBM'in 360 ana bilgisayarı

pazara girdikten sonra, savunma sanayisinde, özel hizmetlerde ve araştırma enstitülerinde kullanılan ticari sistemler ortaya çıkmaya başlamıştır [44].

80'lerin başından itibaren, KDS'nin alt sınıfları oluşmaya başlamıştır. Örnek verecek olursak: Yönetim Bilgi Sistemi, YBS, GKDS, OKDS vb.... Temel olarak, bu sistemleri bir çatı (framework) olarak tanımlayabiliriz. Hiyerarşinin çeşitli seviyelerinde (bireyden şirkete) verilerle çalışabilen ve içine herhangi bir algoritma/mantık uygulanabilecek yapıdadırlar. Buna örnek olarak, Texas Instruments tarafından United Airlines için geliştirilen ve Saha Operasyonlarında karar vermeyi destekleyen - kapıların atanması, optimal park zamanının belirlenmesi vb. GADS (Gate Assignment Display System)'dir [43].

1980'lerin sonunda, "what-if" analizine izin veren ve daha gelişmiş modelleme araçlarını kullanan gelişmiş KDS sistemleri ortaya çıkmaya başlamıştır [45].

Son olarak, 90'lı yılların ortalarından itibaren dünyada, istatistik ve makine öğrenmesi, oyun teorisi ve diğer karmaşık modelleme araçlarına dayanan KDS sistemleri görülmeye başlandı [46].

2.4 KDS Sınıflandırılması

KDS, bilgisayar tabanlı bir otomatik sistemdir. Asıl görevi, verileri eksiksiz analiz ederek zor şart/koşullarda karar vermeye yardımcı olmaktır. Veri tabanı yönetim sistemleri ve yönetim bilgi sistemleri birleştiğinde KDS ortaya çıkar. KDS çok sayıda seçim kriterleri olan karmaşık bir bilgi ortamında karar vermeyi desteklemek için kullanılır.

Çok sayıda seçim kriterleri, nihai kararların sonucu tek bir kritere değil de temelde birçok kritere dayanarak elde edildiğini gösterir. Bilgi karmaşıklığı, büyük miktarda veri işlemeyi gerektirmesinden kaynaklanmaktadır. Bu işlemi modern bilgisayar teknolojisi olmadan gerçekleştirmek zor ve neredeyse imkansızdır.

KDS iki ana görevi yerine getirir:

- Optimizasyon - mümkün olan en iyi çözümü seçmek.
- Sıralama - tercihe göre bir dizi olası çözüm oluşturmak.

2.4.1 KDS Çeşitliği

Literatürde KDS farklı şekilde sınıflandırılabilir. [47]'e göre en popüler 3 sınıflandırma aşağıda verilmiştir.

1. Uygulama alanına göre;

- İşletme ve yönetim (fiyatlandırma, işçilik, ürünler, strateji vb.).
- Mühendislik (ürün tasarımı, kalite kontrol vb.).
- Finans (krediler ve borçlanmalar).
- Tıp (ilaçlar, tedaviler, tanı).
- Çevre.

2. Veri / model açısından (Stephen Alter tekniği);

- Gerekli verilere erişim sağlayan sistemler (File Drawer Systems).
- Hızlı veri işleme için sistemler (Data Analysis Systems).
- Gerekli çözüm türüne göre veri erişim sistemleri (Analysis Information Systems).
- Finansal sonuçları hesaplama sistemleri (Accounting & Financial models (systems)).
- Simülasyon sistemleri (Representation models (systems)).
- Optimizasyon problemlerini çözen sistemler (Optimization models (systems)).
- Kural tabanlı çıkarım sistemleri (Suggestion models (systems)).

3. Kullanılan araç tipine göre

- Modele dayalı KDS (Model Driven) - Problem çözmede veya karar vermede yardımcı olmak için bir model kullanır. Bir model istatistiksel, finansal, matematiksel, analitik, simülasyon veya optimizasyon olabilir. Modele dayalı bir KDS, kullanıcılarının özel gereksinimlerine bağlı olarak tek bir model veya birden fazla modelin bir kombinasyonunu kullanabilir. Basit modeller temel işlevsellik sağlarken, iki veya daha fazla modelin kombinasyonu kullanıcıların karmaşık verileri analiz etmelerini sağlar. Modele dayalı KDS genellikle veri yoğun değildir. Aksine karar vericilerin

girdiği parametreleri kullanır ve durumu analiz etmelerine yardımcı olurlar.

- Veri odaklı KDS (Data Driven) - Bu sistemler, büyük organizasyonel sistemlerde bulunan "veri havuzlarını" analiz eder ve veriye gömülü olan yararlı bilgileri çıkarmalarını sağlayarak karar vermeyi destekler. Bir veya birden fazla veritabanlarına erişimi vardır.
- İletişime odaklı KDS (Communication Driven) - İletişimi, işbirliğini ve paylaşılan karar verme desteğini vurgulayan bir KDS türüdür. İletişime odaklı grup KDS, her katılımcının fiziksel olarak buluşmadan, bulundukları yerlerden gerçek zamanlı olarak iletişim göndermesini, almasını ve başkalarıyla etkileşimde bulunmasını kolaylaştırır.
- Belgeye odaklı KDS (Document Driven) – Yapılandırılmamış belgelerin alınması ve yönetilmesine odaklanır; endeksli dizi belge deposudur (genellikle çok boyutlu). Bu sistemler, eksiksiz belge alımı ve analizi sağlamak için çeşitli depolama ve işleme teknolojilerini bütünleştirir.
- Bilgiye odaklı KDS (Knowledge Driven) - Belirli karar verme kategorileri hakkında uzman bilgisinden yararlanmak için tasarlanmıştır. Kararlar benzer sorun yapılarına dayanan gerçekler, kurallar ve prosedürlerle desteklenir.

KDS'nin ortak noktası, elde edilen sonuçların değerlendirilmesi ve sıralanması için kullanılacak dizi kriterlerin belirlenmesidir. Kullanıcıya doğru seçim yapmada KDS yardımcı olur.

Analiz yapılması ve sonuçların çıkarılması için farklı yöntemler kullanılır. Genel olarak tanımlayacak olursak:

- Sinir ağları.
- Durum analizi.
- Kümeleme analizi.
- Örnek havuzu (base of precedents),
- Evrimsel hesaplama ve genetik algoritmalar.
- Bilgi arama.
- Bulanık mantık.

- Veri madenciliği.
- Karar ağaçları.
- Veri tabanlarında bilgi aramak.
- Simülasyon.
- Bilişsel modelleme, vb.

Listelenen yöntemlerin çoğu yapay zekâdan ödünç alınmıştır. Karar verme sürecinde yapay zekâ tekniklerine ve metotlarına, bulanık mantığa, genetik algoritmalarına, sinir ağlarına vb. yöntemlere dayanan sistemlere Akıllı Karar Destek Sistemleri denir. Akıllı karar destek sistemleri, yapay zekâ tekniklerini kullanarak karmaşık, kesin olmayan ve kötü yapılandırılmış sorunları çözmek için kurumlardaki karar vericilere destek vermek için veri, uzman bilgisi ve modelleri kullanan etkileşimli bilgisayar tabanlı sistemlerdir. KDS yöntemleri yapay zekâ yöntemleri olduğunda akıllı hale gelir. KDS, yönetim kontrolü ve stratejik planlama gibi işlerde, karar alma sürecinin fikir toplama, alternatif geliştirme ve karar verme aşamalarında kullanılabilir. Bu sistemler, şirketin kimlik bilgilerine dayanarak stratejik yönetim görevlerinde karar alma imkânı sunar.

KDS: Veri analizi, modelleme, öngörme ve karar verme için kullanılan bir dizi yazılım aracıdır. Şirketin kendi tasarımlarından ve edinilen yazılım ürünlerinden (Oracle, IBM, Cognos) oluşur.

Haettenschwiler, kullanıcı ile olan ilişkiyi kullanarak pasif, aktif ve işbirliğine dayalı KDS'yi ayırt eder [49]:

- Pasif - karar alma sürecinde destek olur, ancak kesin bir çözüm öneremezler;
- Aktif - doğru kararın oluşturulmasında doğrudan yer alırlar;
- İşbirliğine dayalı/Kooperatif – tüketici ile etkileşimi içerir. Tüketici, sistem tarafından önerilen çözümü değiştirebilir, ekleme yapabilir ve doğrulayabilir, ardından sonucu kontrol etmek için tekrar sisteme aktarma hakkına sahiptir. Karar, sonucu kabul edene kadar tüketiciye tekrar iletilir.

Bu kadar sınıflandırma çeşitlerine rağmen, KDS'nin gereklilikleri ve nitelikleri 4 bölüm içerisinde tanımlanır [50]:

1. Kalite.
2. Organizasyon.
3. Kısıtlamalar.
4. Model.

Aşağıdaki tabloda hangi gereksinimlerin, hangi bölüm altında yerleşeceği gösterilmiştir:

Tablo 2.3 KDS Gereksinimleri

Sistem kalitesi	Doğruluk, Tepki hızı, Yorumlanabilir olması
Mühendislik	Ölçeklenebilirlik, sistemin kompaktlığı, esneklik, kullanılabilirlik, sistem kurulabilirliği
Kaynakların kalitesi	Veri karmaşıklığına toleranslı olması, öğrenme eğrisi, veri gürültüsü toleransı, seyrek/net olmayan verilere toleranslı olması
Fiziksel kısıtlamalar	Geliştirme hızı, uzmanlardan bağımsızlık, hesaplama kolaylığı

Ayrıca, ölçeklenebilirlik (çevik yaklaşımda), kötü verileri işleme yeteneği, kullanılabilirlik ve kullanıcı dostu arayüz ve kaynakların basitleştirilmesi gibi önemli özelliklerin de vurgulanması gereklidir.

KDS kullanılan Veri Madenciliği Araçları 4 gruba ayrılır [51, 52, 53, 54]:

Tablo 2.4 Veri Madenciliği Yöntemleri ve Araçları

Yöntemler	**Kaynaklar**
İstatistiksel veri işleme yöntemleri – İstatistiksel verilerin analizi; – İlişkilerin ve modellerin belirlenmesi; – Çok değişkenli istatistiksel analiz; – Dinamik modeller, zaman serileri analizi ve model tahmini	BMDP ,Data-Desk, SAS, Statistica, SPSS, Systat, Statgraphics, S-Plus, Scenario(BI), TimeLab
Sibernetik yöntemler, kendini geliştiren sistemler ilkelerine dayanarak - sinir ağları yöntemleri, evrimsel ve genetik programlama.	NeuralWorks Pro, NeuroSolution, PolyAnalyst, NeuroShell, GeneHunter, BrainMaker, OWL, 4Thought (BI), Statistica Neural Networks, MatLab Neural Network Toolbox

Optimizasyon problemlerini çözmek için geleneksel yöntemler: varyasyonel yöntemler, Çeşitli matematiksel programlama yöntemleri;	Math CAD, MatLab
Uzman yöntemler: – "En yakın komşu" yöntemi; – Karar ağacı yöntemi; – Sabit matematiksel modellere dayalı, konuya yönelik durum ve tahmin analiz sistemleri; – Veri görselleştirme yöntemleri.	Pattern Recognition Workbench, KATE tools, IDIS, C5.0 и SIPINA, Street Money, METASTOCK, SuperCharts,Candlestick Forecaster Mineset, Impromptu (BI)

2.5 Uzman Sistemler

Uzman Sistemler (US), verilen bilgiyi belirli problemler için önceden belirlenmiş kurallara istinaden analiz edip, kullanıcıya sonuç veya çözüm sunan bilgisayar tabanlı sistemlerdir. US'yi diğer KDS'lerden ayırt eden özellik: US belli bir alanda uzman olan kişilerin uzmanlıklarına dayanarak çözüm arar. İyi tasarlanmış sistemler, belirli problemlerin çözümünde uzman insanların düşünme işlemlerini taklit ederler. Burada uzman sistem tabiri kullanılmasının sebebi, sistemin bir veya daha fazla uzmanın bilgilerine sahip olarak onun veya onların yerini almaya yönelmesinden dolayıdır. US'in amacı, bilgi teknolojilerinin en son trendlerinden suni zekâ ve uzman sistemlerin tanıtımının yapılması ve uygulanması konusunda tartışma ortamı oluşturmak. US'in oluşturulmasında, sırasıyla; tanımlama, kavramsallaştırma, formüle etme (yazılım), test etme ve değerlendirme aşamaları uygulanır. Bir US; kural tabanı, veri tabanı ve kural çözümleyici olmak üzere üç bölümden oluşur.

US, uzman veya bilgi sahibi insan kısıdının olduğu durumlarda (insan faktörü), çok çalışma ve stres gibi faktörlerden etkilenmemeleri nedeniyle tutarlı ve düşük maliyetli birer çözüm olma gibi faydası vardır. Genellikle, alternatif kararları keşfetmek ve muhakeme süreçlerinin açıklamalarını sağlamak için daha uygundurlar.

KDS yapılandırılmamış karmaşık problemlerin analizine imkan sağlayarak yönetim biliminin sınırlarını genişletmiştir. Uzman sistemler ise ancak bir uzman insanın çözebileceği karmaşık problemlerin çözümüne olanak sağlamaktadır.

KDS geliştirirken öncelikli olarak ilgili sistem kimin tarafından, ne zaman, ne kadar sıklıkla kullanılacağı araştırılmalı, analiz ve kontrol edilmelidir. KDS'de yapılan değişiklik ve yeniliklere ait konfigürasyon yönetimi oluşturulmalı, yapılan değişimler tüm kullanıcılara bildirilmelidir. Belirli aralıklarla KDS geliştiricilerinin ve kullanıcılarının katılım sağlayacağı toplantıların düzenlenmesi gerekir.

2.6 KDS Geliştirirken Kullanılan Yöntemler

KDS'lerde en çok kullanılan yöntem ve algoritmalar: Karar ağaçları, YSA, bulanık mantık, genetik algoritmalar, oyun teorisi, olasılık teorisi, graf teorisi vb. sıralanabilir. Aşağıdaki bölümlerde bu yöntem ve algoritmalar kısaca açıklanmıştır.

2.6.1 Karar Ağaçları

KDS'de en çok kullanılan algoritmalardan birisi karar ağaçlarıdır. Karmaşık bir soruna açıklık getirmek ve cevap bulmak için karar ağaçları kullanılır. Karar ağaçlarının yapısı; kullanıcıların birçok olası problemi çözmelerini ve bu çözümleri basit bir şekilde görüntülemelerini; farklı olaylar veya kararlar arasındaki ilişkiyi de basit, anlaşılması kolay bir formatta göstermelerini sağlar. Ağacın en uzağındaki dalları nihai sonuçları temsil eder.

Karar Ağaçları, şartlı olasılıklarla yaratılan ve çıktıları (sonuçları, maliyetleri, olayları vb.) ağaç şeklinde gösterilebilen bir sınıflandırma yöntemdir. Karar ağacı, kök düğüm, dallar ve yapraklardan oluşmaktadır. Yapraklar sınıflandırmanın oluştuğu yerlerdir, dallar ise her bir olayın sonucunu içermektedir. Sınıflandırma kuralları, kök düğümden yaprak düğümlere doğru giden yollardır [65].

Avantajları:

- Karar ağacı oluşturmak zahmetsizdir.
- Küçük ağaçları yorumlamak kolaydır.
- Anlaşılabilir kurallar oluşturulabilinir.
- Sürekli ve ayrık nitelik değerleri için kullanılabilir.
- Görsel değerlendirme olanağı sağlar.
- İş zekâsı (Business Intelligence) için ideal yöntemdir.
- Niteliksel ve niceliksel veri setini destekliyor olmasıdır.

Dezavantajları:

- Sürekli nitelik değerlerini tahmin etmekte çok başarılı değildir.
- Sınıf sayısı fazla ve öğrenme kümesi örnekleri sayısı az olduğunda model oluşturma çok başarılı değildir.
- Zaman ve yer karmaşıklığı öğrenme kümesi örnekleri sayısına, nitelik sayısına ve oluşan ağacın yapısına bağlıdır.
- Hem ağaç oluşturma karmaşıklığı hem de ağaç budama karmaşıklığı fazladır.

2.6.2 Genetik Algoritmalar

Genetik algoritma (GA), 1960 yılında John Holland tarafından tanıtıldı [83]. Genetik algoritma, Darwin'in evrim teorisine benzetilerek rastgele bir arama yöntemi uygular. GA doğal seleksiyona dayanır (en güçlülerin hayatta kalma ilkesine göre çalışan temel evrim mekanizmasıdır). Bu süreci taklit eden genetik algoritma, problemin optimal çözümlerine yakın çözümler bulmayı mümkün kılar. Genetik algoritmasında, soruna ait olası çözümlerin bir dizisi mevcuttur. Bu olası çözümlerin arasından en iyi olası çözüm bulunur. Her adımda seçilen en iyi çözümler çaprazlama ve mutasyon işlemine tabi tutulur. Bu şekilde elde edilen yeni çözümler en iyi çözüme ulaşılana kadar işlem birkaç kez tekrar edilir.

Algoritmanın avantajları:

- Diğer arama algoritmalara göre daha hızlı sonuç üretir;
- Karmaşık problemleri çözmek için kullanılır;
- Büyük ve düzensiz veri içeren problemler çözümünde diğer algoritmalara göre avantajlıdır;
- İyi tanımlanamayan problem çözümlerinde iyi sonuç verir;
- Diğer arama algoritmalarına göre daha hızlı çalışır ve bilgisayarın performansını yavaşlatmaz.

Algoritmanın dezavantajları:

- Genetik algoritma bulunan çözümün optimal olacağını garanti etmiyor (genellikle “kısa sürede” sorunun “makul derecede iyi” bir çözümünü bulmak için kullanılır);

- Problemin tasarım zorluğu (genetik algoritma formuna dönüştürmek zordur).

2.6.3 Bulanık Mantık

Bulanık kümeler ve bulanık mantık teorisi, matematiksel dilsel terimlerin işlenmesini sağlamak için geliştirilmiştir. Bulanık modeller karar vermeye yardımcı olurlar. Bulanık mantık yöntemi ile belirsizlik ortamlarında yapılan işlemlerinin çözülmesi etkin, daha az zaman ve maliyet ile karar verebilmeyi sağlar**.** Bu yöntem daha çok tahmin ve sınıflandırma problemlerinin çözümünde kullanılır. Tümdengelim ve tümevarım ile sonuca varmada, başka deyişle, mantıksal sonuçlar elde eden sistemlerde de kullanılır [55].

Bulanık mantığın uygulanacağı sistemin veya problemin, karmaşıklığı ne kadar fazla ise ve kararı sağlayacak veri miktarı ne kadar yetersiz ise bulanıklık etkisi o kadar fazla olacaktır. Klasik mantıkla (0 ve 1 mantığı), durumlar veya nesneler bir kümeye veya sınıfa aittir. Klasik küme teorisinde olduğu gibi, kümeye dahil olan elemanlar üyelik değerleri 1, kümeye dahil olmayan elemanların üyelik değerleri ise 0 olarak ifade edilir. Bulanık kümeye dahil olup olmadıkları belirsiz olan elemanların üyelik değerleri ise 0 ile 1 arasında değer alabilirler.

Bulanık mantık insan düşünce yöntemine yakın olmasından ötürü anlaşılması kolaydır. Geniş ve kapsamlı analizlerden uzak ve sezgisel yaklaşımı olduğu için basit ve esnek yapılıdır, deneyimleme yoluyla kendini sürekli geliştirir. Daha çok zor problemlerin çözümlenmesinde tercih edilir. Teorinin sahibi Zadeh 1965 yılında yayınlamış makalesinde: İnsanların bulanık durumlardaki karar mekanizmalarının, eksik veya tamamen güvenilir olmayan bir bilgiye dayanan bir sorunun yaklaşık bir cevabını çıkarabilme yeteneklerine bağlı olduğunu söyler. "Neredeyse her problem için bulanık mantık olmadan çözüme gidebilirsiniz, ancak onu kullanırsanız daha hızlı ve ucuz yolu seçmiş olursunuz." demiştir [66]. Bugün, bulanık mantık, belirsizliklerin anlatımı ve belirsizliklerle çalışılabilme olanağı olarak tanımlanır.

Avantajları [56]:

- İnsanın düşünce sistemine ve tarzına yakındır.

- Karmaşık verinin analizi ve modellenmesinde meydana gelebilecek farklılıkları azaltır.
- İnsan tecrübesi ile elde edilen nitel kuralların birleştirilmesi için uygundur.
- Esnektir ve kurallarda değişiklik yapılmasına izin verir.
- Kesin olmayan ve hatalı giriş bilgileri sistem tarafından kabul edilir.
- Uygulamasında matematiksel bir modele gereksinim duymaz.
- Sistemler kolayca inşa edilebilir.
- Bu sistemler insanın akıl yürütme ve karar verme becerisini içerdiğinden, farklı uygulamalarda karmaşık çözümlere çözümler sağlamada yararlıdırlar.
- Belirsizlik problemleri ile başa çıkmanıza yardımcı olur.
- Sistem performansını iyileştirmek veya değiştirmek için kolayca değiştirilebilir.
- Diğer yapay zekâ yöntemleri ile uyumludur.
- Karmaşık sorunlara en etkili çözümü sunar.

Dezavantajları:

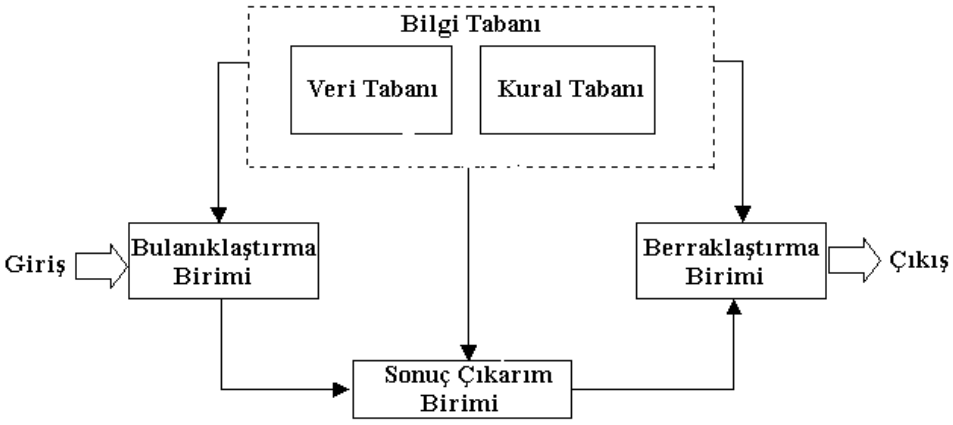
- Bulanık denetimde kullanılan kurallar deneyime bağlıdır.
- Üyelik fonksiyonlarının seçiminde belirli bir yöntem yoktur. En uygun fonksiyon, deneme yolu ile bulunur. Bu da uzun bir zaman alır.
- Denetlenen sistemin bir kararlılık analizi yapılamaz ve sistemin nasıl cevap vereceği önceden kestirilemez. Yapılacak tek şey benzetim çalışmasıdır.
- Bulanık mantık her zaman doğru sonuçları vermez, sonuçlar varsayıma dayalıdır.

Bulanık mantık genellikle bilinen metot veya yöntemlerle sonuç alınamayan sistemlerde kullanılır. Bulanık mantık yaklaşımı makinelere insanların özel verilerini işleyebilme ve onların deneyimlerinden ve önsezilerinden yararlanarak çalışabilme yeteneği verir.

Bulanık Mantık Kontrol sistemi dört ana yapı içerir.

- Bulanıklaştırma.
- Bilgi Tabanı (veri tabanı, kural tabanı).

- Çıkarım.
- Durulaştırma.



Şekil 2.7 Bulanık Kontrol Sisteminin Yapısı

Bulanıklaştırma birimine giren bilgiler bulanık bir büyüklük formuna getirilir. Çıkarım biriminde bulanık kümeler bulanık eğer-ise kuralları ile birleştirilir. Durulaştırma kısmında, çıkarım biriminden elde edilen sonuçlar durulaştırıcı birimine gönderilerek net bir çıktı üretilir, sonuç tekrar tam değere dönüştürülür. Bulanık mantık ile kontrol konusundaki ilk uygulama 1974'de Mamdani tarafından buhar makinesinin kontrolü ile gerçekleştirilmiştir [76]. Mamdani çıkarım sistemi, bulanık küme teorisi ile inşa edilmiş olan ilk kontrol sistemi olarak bilinir. Minimum korelasyonu veya Mamdani önermesinin formülü aşağıdaki gibidir:

$$\mu_R(x, y) = \min[\mu_A(x), \mu_B(y)] \tag{2.1}$$

Bu denklemde, R= A×B boyutlu bulanık A ve bulanık B kümelerinin bulanık Kartezyen kümesidir. $\mu_A(x) \geq 0{,}5$ ve $\mu_B(y) \geq 0{,}5$ için klasik mantık önerme yaklaşımının, Mamdani önermesine indirgenmiş halini ifade eder [77]. Mamdani çıkarımında, "eğer-ise" kuralının sonucu bir bulanık kümeyle tanımlanır. Her bir kuralın bulanık çıktı kümesi, sistemde eşleşen bir sayı ile yeniden şekillendirilir ve bu yeniden şekillendirme sonucunda elde edilen bulanık kümelerin tamamı toplanıp, sonrasında durulaştırılır [78].

Bulanık mantık havacılık sektörü içinde de yaygın olarak ta kullanılır. Örnek olarak; Hava sahasında yükseklik kontrolü, helikopter modelinin kontrolü, uydu uzaklık kontrolü, uçak içi hava ve basınç kontrolü, elektronik cihazların kontrolü, insansız havaaraçların ve helikopterlerin kontrolünde de kullanılır.

2.6.4 YSA

YSA, basit ve birbirlerine paralel ve dağınık olarak bağlı ve her biri kendi belleğine sahip çok sayıda yapay nöron işlemcilerden oluşur. YSA zaman zaman; sinirsel hesaplama (neural computation), ağ hesaplama (network computation), bağlantılı ağlar (connectionist networks), paralel dağıtılmış ağlar (parallel distributed networks) ve nöromorfik sistemler (neuromorfic systems) ifadeleriyle de anılır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır. Bir nöronun matematiksel bir modeli şu şekilde temsil edilebilir: Özelliklerini özelleştirebilen ve değiştirebilen, doğrusal olmayan elemandır. YSA'nın en önemli üstünlükleri, öğrenme kabiliyeti olması ve farklı öğrenme algoritmaları kullanabilmesidir. Bulundukları ortama uyum sağlar, adaptedir, eksik bilgiyle çalışabilir, belirsizlikler altında karar verebilir, hatalara karşı toleranslı olurlar. Bunun yanı sıra en sık belirtilen sakıncaları ise sistemin çalışmasının analiz edilememesi ve öğrenme işleminde başarılı olamama riskidir. YSA genellikle sınıflandırma problemlerinin çözümünde kullanılır. YSA'nın "eğitilmesi" gerekir, bunun başlangıç değerleri ve elde edilen sonuç değerlerinin bilinmesi gerekir. Bağlantı ağırlıkların güncellenmesi diğer nöronların "eğitilmesine" yardımcı olur ve eğitilen nöron problemlerini çözebilir hale getirir. Kullanıcının sınıflandırma kurallarını değiştiremediğini dikkate almak gerekir. YSA yapay zekânın bir bileşenidir ve içinde sinyal işleme için canlıların nöronlarında meydana gelen benzer fenomenleri kullanır. YSA, bazı fonksiyonlara dayanarak, giriş ve çıkış bilgisi ile eşleşen bir sistemdir. Bu işlev, eğitim aşamasında ağ tarafından algılanır ve kullanıcı tarafından bilinmeyen olarak kalır. Ağın en önemli özelliği, sinir ağının tüm bağlantıları ile bir defada bilginin paralel işlenmesi olasılığıdır. Bu bilginin işlemesini önemli ölçüde hızlandırır. Eşit derecede önemli bir diğer özellik, biriken bilgiyi öğrenme ve özetleme yeteneğidir. Öğrenme sürecinde sinir ağı neyin normal (doğru) olduğunu ve neyin anormal (yanlış) olduğunu öğrenir. Bu nedenle, bir veri iletim ağının durumunu teşhis etmek için bir sinir ağını kullanmadaki en önemli görev, bir sinir ağını tüm anormal ağ koşullarını doğru şekilde tanımak için eğitmektir [57].

YSA kullanımı, sistemlerin aşağıdaki yararlı özellikleri sağlar [58]:

- Bilinmeyen veri ortamında problem çözümü sağlar. Çok sayıda örnekle çalışması sinir ağına, durum yasalarının girdi ve çıktı verileri arasındaki ilişkisinin bilinmediği durumlarda sorunları çözme yeteneği verir.
- Giriş verilerinde gürültüye (problem çözümünde kullanılmayacak veriler) karşı direnç sağlar. Sinir ağının kendisi, hangi sinyallerin önemsiz/bilgisiz olduğunu, sorunu çözmek için gürültülü olduğunu ve onları dışarıda bıraktığını belirler.
- Çevresel değişikliklere uyum sağlama.

YSA çevresel değişikliklere uyum sağlar. Örneğin, belirli bir ortamda çalışmak üzere eğitilmiş sinir ağları, çevresel parametrelerdeki küçük dalgalanma koşullarında çalışmak için de kolayca eğitilebilirler. KDS ima ve çıkarım kurallarını kullanırken, sinir ağları öğrenme yeteneğine sahiptir. KDS ile sinir ağlarının birleşimi olan sistemde, bilgi tabanı yerine eğitimli bir sinir ağı kullanılır. Bilgiye dayalı KDS'den farklı olarak, bu tür bir sistem güvenirliği kesin olmayan ve eksik verilerle çalışabilir. Sorunlu alan bilgisi ağ eğitiminde kullanılabilir ve eğitimden sonra sinir ağları IF-TO - bilgi tabanı için bir dizi kuralın rolünü üstlenir. Böylece, sinir ağlarının ve uzman sistemlerin etkileşimi, bu teşhis modellerinin dezavantajlarını en aza indirir ve bu yaklaşımların kombinasyonu diğerlerine göre avantaj sağlar.

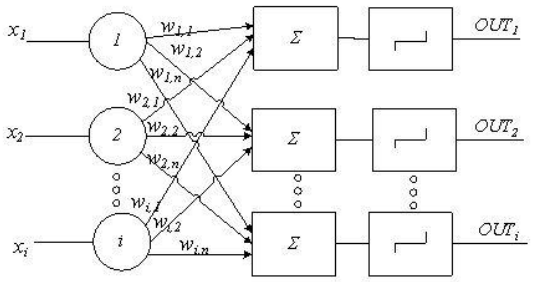
YSA, mevcut örneklerin ilişkilerini değerlendirerek ve elde ettiği bilgileri kullanarak karar veren sistemlerdir [59]. YSA olasılık işlevlerini yerine getirmek için kullanılan bir sistemdir. YSA değişen girdilere uyum sağlayabilir. Böylece ağ, çıktı ölçütünü yeniden tasarlamaya gerek kalmadan mümkün olan en iyi sonucu üretir. YSA, insan beynindeki nöronların çalışmasını taklit ederek geliştirilmiş bir donanım ve/veya yazılım sistemidir.

YSA, basit ve birbirine bağlı basit nöronlardan oluşan bir sistemdir (yapay nöronlar). Bu tür nöronlar genellikle basittir (özellikle kişisel bilgisayarlarda kullanılan nöronlar karşılaştırıldığında). Benzer bir ağın her nöronu, yalnızca düzenli olarak aldığı sinyallerle ilgilenir ve periyodik olarak diğer nöronlara sinyaller gönderir [60]. Bununla birlikte, kontrollü bir etkileşime sahip oldukça

büyük bir ağa bağlı olduklarında, bu tür basit nöronlar karmaşık görevleri gerçekleştirme yeteneğine sahiptir.

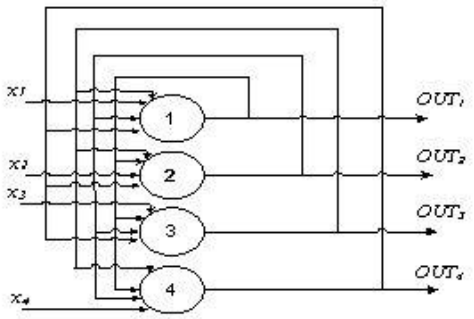
Amerikan nörofizyoloğu Francis Rosenblatt (1958), kendi çalışmalarında, biyolojik sinir ağlarındaki bağlantıların tesadüfen oluştuğunu belirtmiştir. Belirli sınıflar arasındaki tesadüfen bağlanan ağların özellik alanını tanımlayan kendi sinir ağı modelini önermiştir. Bu modele "Perceptron" öğrenme kuramı demiştir [41].

Bu ve benzerleri sistemlere, algılayıcı adı verilir ve esas olarak çoklu girdilere sahip ağırlıklar vasıtasıyla bağlanan tek bir yapay nöron tabakasından oluşur.



Şekil 2.8 Tek Katmanlı Algılayıcı

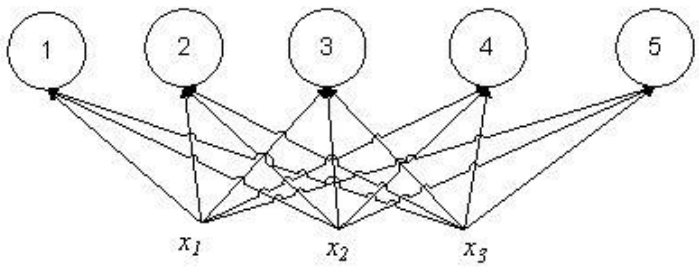
Sinirsel hesaplamanın modern matematiksel modellemesi 1982'de Hopfield'in çalışmasıyla başladı. Hopfield, sadece karar üreten beyinleri modellemek değil faydalı sistemler üretmek üzere bir yaklaşımı anlatmıştır. Yazdığı makalede, açıklık ve matematik analizi ile yapay sinir ağların nasıl çalıştığını ve neler yapabileceklerini göstermiştir [60]. Hopfield modelinin dezavantajı, enerji fonksiyonunun küresel minimumundan ziyade lokalde stabilize olma eğilimleridir. Dolayısıyla Boltzmann makinesi, kombinasyonel optimizasyon problemlerini ve yapay zekâ problemlerini çözmek için modelin evrimsel bir gelişimidir [42].



Şekil 2.9 Dört Nöronlu Hopfield Ağı

1940’ların sonlarında psikolog Donald Hebb, Hebbian öğrenme olarak bilinen hipotezi oluşturmuştur. Çalışmalarında sinir hücresini kullanan Hebb iki sinir hücresi arasındaki etkileşimleri araştırarak ortaya sunduğu kurallar sinir ağlarının temelini oluşturmuştur. Hebb kuralında, eğer iki nöron birbirlerinden bilgi alıyorsa ve ikisi de aktifse o iki nöron arasındaki bağın kuvvetlenmesi gerekir. Yani aktif olan nöron bağlantılı olduğu nöronu da aktif hale getirmeye çalışacaktır. Hebb kuralı eğitimsiz bir öğrenme kuralıdır [76].

1980'lerde Kohonen tarafından geliştirilen kendi kendini düzenleyen özellik haritası (SOFM ağı - Self-Organizing Feature Map), sayı eğitim vektörlerinin boyutuna karşılık gelen bir dizi girdi öğesine ve prototip görevi gören bir dizi çıktı öğesine sahiptir. SOFM ağının temel mimarisi, Şekil 2.10'te gösterilmiştir.



Şekil 2.10 Üç Giriş ve Beş küme Elemanına Sahip Kohonen Ağı

1980'lerin sonlarında, tek katmanlı ağların yeteneklerini aşan yayılma karşıtı ağlar geliştirildi. Hatalı (geri) yayılım gösteren ağlara kıyasla, bunların eğitim süresi yüz kat azaltılabilir. Geri yayılım ağları (geri beslemeli), uzun bir öğrenme prosedürünün mümkün olmadığı uygulamalarda bir çözüm sağlayabilir. İleri beslemeli ağlar, iyi bilinen iki algoritmayı birleştirir: Kohonen özdüzenleyici özellik haritası ve Grossberg yıldızı. Bu kombinasyon, hiçbirinin ayrı ayrı sahip olmadığı özelliklere yol açar.

Böylece, karmaşık bir yönetim problemine ilişkin temel bilgilerin analiz edilmesi, öngörülmesi, sentezlenmesi ile ilgili entelektüel yeteneklerin tam olarak uygulanması ve sonuçta doğru kararların alınması, modern ve gelecekteki akıllı karar destek sistemleri ve akıllı sistemlerinin ortaya çıkmasıyla mümkün olabilmektedir. Akıllı sistemler, belirli bir duruma ve göreve uyum sağlayabilen dinamik veri modellerine, yapay zekâ kavramlarına, çözme sürecinde büyük

miktarda verinin paralel işlenmesine ve makul sonuç çıktısı yöntemlerine dayanır. Bu nedenle, akıllı karar destek sistemleri, veri madenciliği sistemleri, kontrol sistemleri ve tahminlerini oluşturmanın en doğru yollarından biridir. Sinir ağlarının, bulanık modellerin ve çok kriterli seçim ve bulanık çıkarım yöntemlerinin teorisi ve pratiğinde modern bilimsel araştırmaların kullanılması yer alır.

YSA'ya dayalı KDS oluşturmak, yalnızca daha önce kullanılan yöntemlerin verimsizliklerinin gösterildiği alanlarda uygundur.

Karar destek sistemlerin tasarım yöntemleri, Tablo 2.5'de gösterilmiştir.

Tablo 2.5 KDS Tasarım Yöntemleri

Yöntem	**Bilgi temsil şekli**	**Uzman sistemlerin (KDS) özellikleri**				
		Bulanık parametrelerin işlenmesi	**Doğrusal olmayan ilişkilerin temsili**	**Değerlendirme**	**Sonuçların güvenilirliği**	**Bilgi edinme yöntemleri**
Olasılık teorisi	Kurallar, ağırlıklar	-	-	+	+/-	-
Olasılık teorisi	Tablolar	-	-	+	-	-
Graf teorisi	Semantik ağlar, Petri ağları	-	-	-	+	+
YSA	Nöronlar: bağlantılar ve ağırlıklar	-/+	+	-	-/+	+
Oyun teorisi	Optimum kriterleri (objektif fonksiyonları)	-	+	-/+	+	+
Bulanık mantık	Bulanık küme	+	+	+	+/-	-
Hibrit KDS (BM + YSA)	Bulanık küme, kurallar	+	+	+	+/-	+

Hibrit KDS (YSAı + BM)	Bulanık nöronlar, bulanık bağlantılar ve ağırlıklar.	+	+	-/+	+/-	+

En iyi sonucu veren hibrit modeller kullanılarak, problem çözmede etkili bir KDS tasarlanabilir.

Bulanık kümeler üyelik fonksiyonlarıyla, t-normları ve k-normları ise sıradan matematiksel işlemlerle tanımlandığından, bulanık mantık yorumları bir sinir ağı olarak gösterilir. Bunun için üyelik fonksiyonlarının nöron aktivasyon fonksiyonu olarak yorumlanması; bir bağlantı olarak sinyal verilmesi; mantıksal t-normları ve k-normlarının matematiksel karşılık gelen işlemleri yapan özel nöron tipleri olarak yorumlanması gerekir. ANFIS (Adaptive Neuro fuzzy Inference System) uyarlanabilir nöro-bulanık çıktı sistemi gibi çok çeşitli bulanık sinir ağları (neuro-fuzzy networks) bulunmaktadır [48].

3
Deneysel Bulgular

İşletmeler için temel gelir kaynağı, onarımı da dâhil olmak üzere filosunda bulunan motor ömürlerin korunmasıdır. Doğru bir şekilde organize edilmiş ve etkin bir şekilde uygulanan onarımlar yalnızca kar ettirmekle kalmaz, aynı zamanda önceden üretilmiş ekipmanların rekabet gücünü de korur. Motorların onarımında, analiz için kullanılan çeşitli istatistik verilere büyük özen gösterilmektedir. Örneğin, tamirden geçen motor arızalarının bir sonucu olarak, bozuk parçalar, montaj birimleri ve bileşenler üzerinde toplanan istatistikler, yeni onarım türlerinin geliştirilmesiyle bozulmayı azaltan ve operasyondaki arızaları ve hataları önleyen çok önemli ve çeşitli analizlerin yapılmasına yardımcı olur. Arızalar ile ilgili istatistiksel verilerin ayrıntılı bir şekilde belirtilmesi (arıza nedeni, motor numarası, toplam çalışma süresi, son onarımdan sonraki çalışma süresi, işletme organizasyonu vb.) analizlere katkıda bulunur. Bu nedenle, en sık arızalanan malzeme kısmının analizi, maliyeti göz önünde bulundurularak, ilk etapta en pahalı parçalar için önlemlerin alınmasını mümkün kılar, çünkü bu parçaların değiştirilmesi tamir maliyetini önemli ölçüde etkiler. Motor parçalarının çalışma süresinin artmasıyla yapılan arızalanma analizi belirli sorunların kaynağını tahmin etmemizi sağlar. Aynı zamanda, motorun temel verilerini etkileyen, testler sırasında elde edilen ana parametreler üzerine istatistiksel bir veri tabanı oluşturulur. Bu tür bir veri tabanın analizi, kritik parametrelere sahip motorların tanımlanmasına olanak tanıyacak ve onarım başlamadan önce, testlerden çıkarılması için uyarıcı önlemler alınabilecektir.

Havayolu firmalarının uluslararası ticari birliği olan Uluslararası Hava Taşımacılığı Birliği, uçak ve sistemleri üreticileri (Boeing, Airbus, Embraer, vb.) ile teknik işbirliği yapan Uluslararası Sivil Havacılık Teşkilatı, hava araçlarının uçuşa hazır olma ve uçuş emniyetini sağlamak için uluslararası bakım ve uçuş regülasyonlarını belirler [74]. Bu regülasyonların uygulanması, ülke ve ülke birliklerinin sivil havacılık kurumları (IACA/Dünya, EASA/Avrupa, BERMUDA/Rusya, FAA/ABD,

SHGM/Türkiye) tarafından yönetilir ve denetlenir [75]. Havayolların ve BOY kuruluşların kendi ülkesinde geçerli olan regülasyonları takip ederek gerekli bakım sürecini yönetmek zorundadır. Tüm regülasyon ve tanımlı iş süreçlerinin eksiksiz ve tam olarak takip edilip uygulandığı yazılımlar vasıtasıyla veya elle işletilen sistemler ile yürütülmektedir.

Hava taşıtlarının bakımı, uçak ve uçak parçalarının tüm çalışma süresi boyunca sürekli hizmet verebilme ve uçuşa hazır olmalarını amaçlayan mühendislik servisleri ve teknik destektir. Bakım uygulanmaz veya eksik yapılır ise, uçak çok hızlı bir şekilde kullanılamaz hale gelir. Uçağın uygun bakımı, uçuş güvenliğinde temel bir faktördür. Uçakların bakımı belli bir süre, saat veya devir sonra havayolları veya BOY kuruluşları tarafından yapılması gereken periyodik muayenelerinden oluşur. Bakım türleri: Transit bakımı, günlük bakım, haftalık bakımı, A-bakımı, B-bakımı, C-bakımı ve D-bakımıdır. A ve B-bakımı basit (kolay) bir bakım iken C ve D-bakımı zor bir bakım şeklidir.

Transit bakım - uçağın her kalkışından önce yapılan en basit bakım şeklidir. Bu kontrol eğer iki uçuş arasında süre üç saatten az ise, uygulanır. İnişten sonraki uçuş eğer üç saatten az bir sürede yapılacaksa uygulanır. Bir sonraki kalkış, inişten üç saatten daha fazla bir süre sonra yapılacaksa, Günlük bakım yapılır.

Günlük bakım - uçağın teknik durumunun günlük kontrolü, 48 saatte bir yapılmalıdır. Bazı durumlarda 96 saat sonra yapılabilir. Genellikle geceleri yapılır.

Haftalık bakım - haftada yaklaşık bir kez yapılır. Hem gündüz hem de gece yapılabilir. Mülkiyet durumu gerektirmez. Genellikle 3-4 saatte tamamlanır.

A-bakımı - ayda bir kez veya her beş yüz saatte bir yapılır: A1, A2, A4, A8 alt kontrollerden oluşur. Sayı arttıkça, iş miktarı da artar. A-bakımı genellikle geceleri havaalanının hangarında yapılır. Bu bakımın içeriği uçağın tipine, devir sayısına veya son kontrolden itibaren uçuş saatlerine bağlıdır. Belli şartlara bağlı olarak kontrol havayolu tarafından ertelenebilir.

B-bakımı - bu kontrol yaklaşık üç ayda bir yapılır. Genellikle geceleri havaalanının hangarında yapılır.

C-bakımı - bu bakım şekli öncekilerden daha karmaşıktır ve her 15 ila 24 ayda veya 7.500 uçuş saatinde bir gerçekleştirilir. C1, C2, C4, C6 ve C8 olarak alt kontrollere bölünmüştür. Bu kontrolü gerçekleştirmek için uçağın bir süre (yaklaşık iki hafta) hizmet dışı kalması gerekir ve büyük bir havaalanı hangarında (çok fazla alan) yapılmasını gerektirir. Bu testin zamanlaması birçok faktöre, özellikle de uçak tipine bağlıdır.

D-bakımı - uçak bakımının en zor şeklidir. Bu kontrol yaklaşık 12 yılda bir yapılır ve 30-40 gün sürer. Bu sırada, uçağın tüm bileşenleri ve parçaları kontrol edilir. Ömrü biten veya testi geçemeyen parçalar değiştirilmelidir. Uçağın bu muayenesi, diğerlerinden daha fazla yer ve zaman gerektirir ve uygun teknik kuruluşta gerçekleştirilir.

Atölye ziyareti (SV-Shop Visit) - uçağın motorlarının bakımı için yapılan zor bir işlemdir. Ortalama 12000 uçuş saatinde bir yapılır.

3.1 KDS Tasarımı

Yaşadığımız dünyada giderek artan bilgi yoğunluğu; birçok verinin kaydedilmesi, sınıflandırılması, anlamlı hale getirilmesi ve bu verileri kullanarak karar vermede kullanılmak üzere geliştirilecek olan yazılımların çeşitlilik gösterilerek artması ihtiyacını doğurmuştur. Aşağıdaki bölümlerde karar verme ihtiyacını karşılamaya yönelik KDS yazılımının tasarlanması aşamaları açıklanmaktadır.

3.1.1 KDS Yapısının Geliştirilmesi

KDS, bilinçli karar verme, zamanında problem çözme ve hızla değişen değişkenler ile problemlerin çözümlenmesi için verimlilik sağlar. Günümüzde her alanda kullanılır ve ihtiyaca göre geliştirilebilir. Temel amaç, belli bir yapıya sahip olmayan problemlere, bilgisayar/sistem desteği ile karar verilebilmektir. Bu genellikle karar vericinin, sürece zorunlu bir şekilde dâhil edilmesini gerektiren etkileşimli bir karar desteğidir. KDS, kullanıcının hızlı bir şekilde çok kriterli problemlerde daha bilinçli ve etkin kararlar vermesini sağlar.

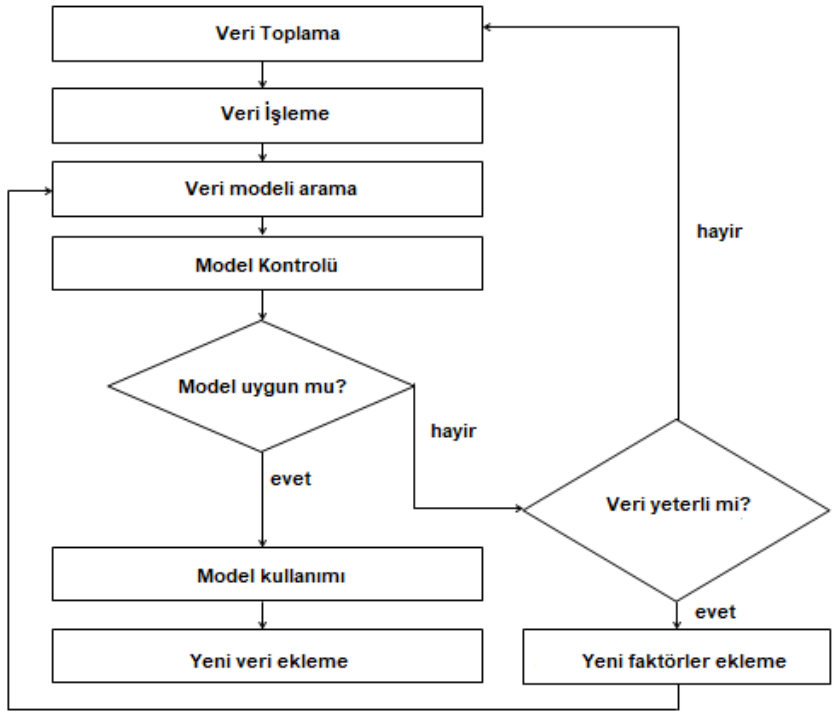
KDS bir yazılım ürünü olarak değerlendirilirse, karar vericiye yardım sağlamak için oluşturulan bir insan-makine kompleksi olarak tanımlanabilir. Genellikle, zayıf formalizasyon ve çeşitli alanlarda zayıf yapılandırılmış verilerle ilgili problemleri

çözmek için kullanılır. KDS, karar verme kurallarını ve veritabanlarını, var olan modellerde kullanan/uygulayan otomatik bir sistem olarak adlandırılır.

KDS ayrıca şunları içerir:

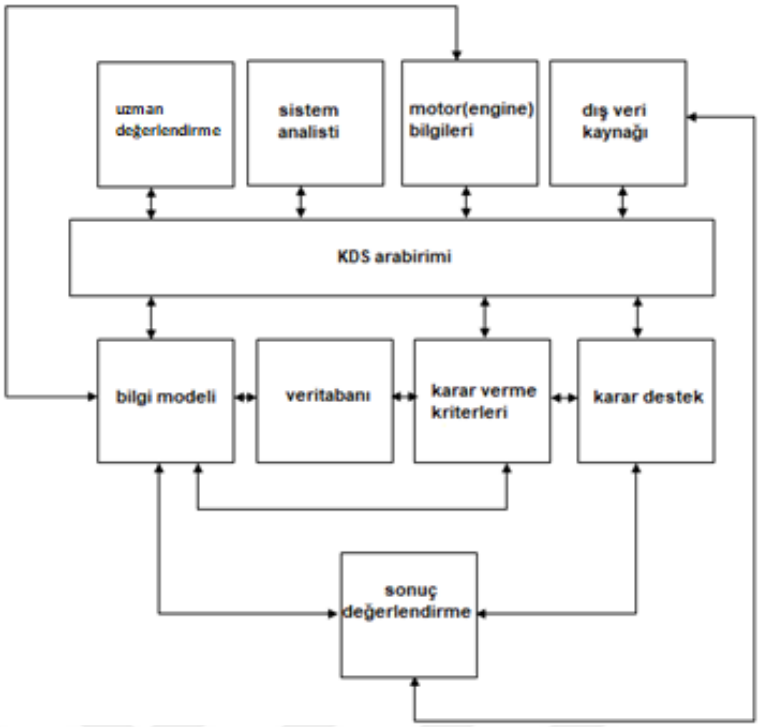
- Karar vericinin, kontrol edilen sistemin durumunu ve bunun üzerindeki etkilerini değerlendirmesine yardım eder.
- Karar vericinin tercihlerinin belirlenmesine destek sağlar.
- Uygun çözümleri üretir.
- Karar vericinin tercihlerine dayanarak olası alternatifleri değerlendirir.
- Alınan kararların (sonuçlarının) analizi ve karar verici tarafından en iyisinin belirlenmesini sağlar.

KDS yapısındaki ana bileşenler:



Şekil 3.1 KDS Ana Aşamaları

Şekil 3.2.'de gösterilen sorunun ana özellikleri yansıtan KDS'nin tasarlanması:

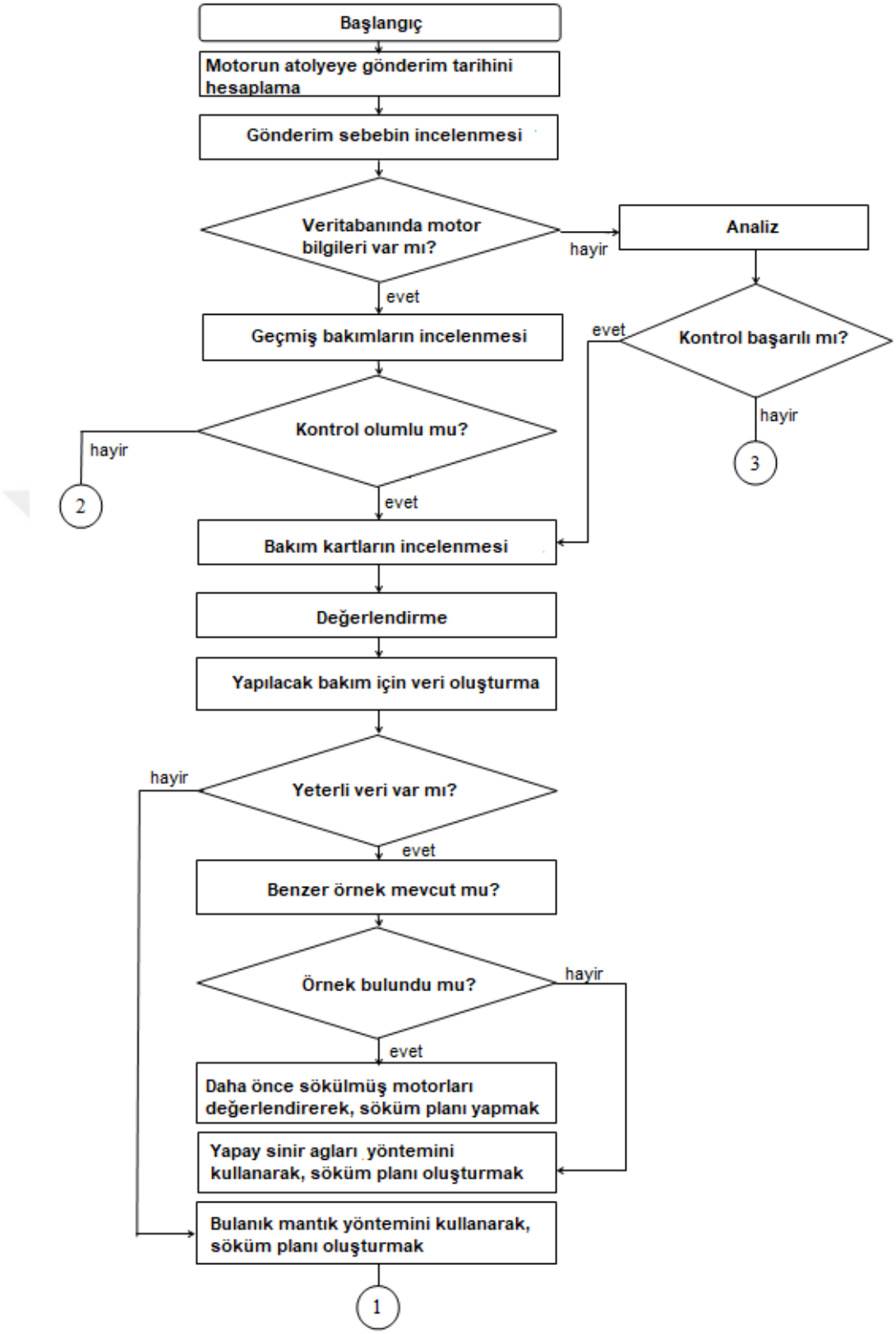


Şekil 3.2 KDS'nin Yapısı

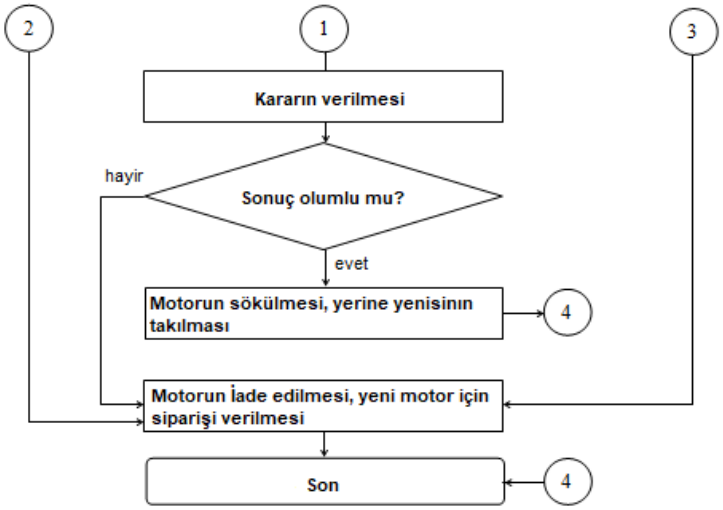
- Motor Bakım Mühendisi (Powerplant Engineer) – Motor takım/söküm sürecini takip eden ve yapılacak işlemlerden sorunlu olan kişi veya departmandır (karar veren kişidir).
- Bilgi modeli –Veriler ile tüm işlemler veri modelinde yapılır; veri toplama, veri analizi, bilgi toplama, bilgi modeli oluşturma, model eğitimi, vb.
- Karar verme kriterleri – Kararların oluşumu, alınan kararın kontrolü ve sistem tarafından verilen kararların analizini içerir.
- Karar destek birimi - Çözüm kümesinin analizini, optimum çözümün seçilmesini ve çözüm kümesinin oluşturulmasını içerir.

3.1.2 KDS Diyagramı

Açıklanan problemin çözümü için YSA ve bulanık mantık yöntemleri kullanarak Şekil 3.5'de verilen KDS diyagramı oluşturulmuştur.



Şekil 3.3 KDS'nin Şeması (a)



Şekil 3.4 KDS'nin Şeması (b)

Motor bakım/söküm planlamasında yer alan önemli kavramlar aşağıda sıralanmıştır. Bu motor takım kavramlarına dayanarak motor bakım/söküm süreci gerçekleştirilir.

3.2 Motor Bakım Kavramları

Motor, uçağın kalbidir. Düzenli nitelikli bakım ve servis gerektirir. En çok kullanılan turbofan uçak motorları:

- CF6 motor ailesi (Airbus A310, A330, Boeing 747, Boeing 767, MD11 uçak tiplerinde kullanılır).
- CF34 Motor Ailesi (Bombardier, Embraer, Comac ARJ21).
- GE90 Motor Ailesi (Boeing 777). Havacılık tarihinin en büyük ve en güçlü turbofan motorudur.
- GEnx Motor Ailesi (Boeing 787, Boeing 747-8F).
- Rolls-Royce Ailesi (Airbus A330, A340, A350, A380, Boeing 777, 787 Dreamliner).

Aşağıda verilen başlıklardan herhangi birinde aksilik veya arıza tespit edildiğinde motorun bakıma/değişime ihtiyacı olduğu çıkarımı yapılabilir.

3.2.1 Motor Basınç Oranı

Motor Basınç Oranı (MBO) - motordaki toplam basınç oranı olarak tanımlanır ve egzozdaki (veya türbin çıkışındaki) toplam basıncın, fan / kompresörün toplam basınca oranı alınarak hesaplanır.

3.2.2 N1 Hız

N1 hız, fanın (veya motor tipine bağlı olarak düşük basınçlı kompresörün) dönme hızıdır ve genellikle tasarım devri yüzdesi olarak sunulur.

3.2.3 Egzoz Gazı Sıcaklığı

Egzoz Gazı Sıcaklığı (EGS) - santigrat derece cinsinden ifade edilir, motor egzozundaki sıcaklıktır ve motorun tasarım seviyesindeki itme üretimindeki verimliliğinin bir ölçüsüdür. EGS, türbin ünitesinden çıkan gazın sıcaklığıdır. EGS ne kadar yüksek olursa motor üzerinde aşınma ve bozulma o kadar fazla olur. Yüksek EGS, düşük motor performansının bir göstergesidir. EGS sınırlarının aşılması, motor parçalarının zarar görmesine ve / veya motor parçalarının ömrünün azalmasına neden olur. Bu bilgiyi göz önünde bulundurarak, EGS'yi mümkün olduğu kadar düşük bir seviyede tutmak önemlidir.

3.2.4 EGS Marjı

EGS Marjı kalkış sırasında ortaya çıkan en yüksek EGS ile sertifikalı 'Redline' (barındırabileceği maksimum rpm değeri) EGS (sivil havacılık her motor tipi için max EGS değerini belirler) arasındaki farktır.

3.2.5 Motorun Sınırlı Ömür Parçaları (Life Limited Part - LLP)

LLP – Motorun ömrü sınırlı parçalarıdır. Motor içerisinde, bu parçalar arıza yaparlarsa motorun sökülerek bakıma gönderilmesi gerekir, çok kritik parçalardır ve uçağın uçuş sayısı ile yönetilir. Çoğu durumda, LLP'lerin beyan edilen ömürleri 15.000 - 30.000 devir arasındadır ve motorun toplam maliyetinin yüksek bir oranını (% 20'den büyük) temsil eder. Bu parçaların ömrü hesaplanarak motor bakıma gönderilir. Ömür takibi iki şekilde yapılır. Birinci yöntemde, ömür takibi parça için belirlenen ömür sınırından uçtuğu uçuş sayısı çıkartılarak kalan ömür elde edilir. İkinci yöntemde ise, ömürlü parçalar genelde 3 modüle ayrılır A, B ve C.

A ile ilk üretildiği tarihten itibaren ömür sayım, B ile ilk bakımdan sonraki ömür sayımı ve C ile ikinci bakımdan sonraki ömür sayımı takip edilir. Ömrün takibi aşağıdaki şekilde gösterilmiştir:

$$Remaining\ A = \left[1 - \left(\frac{Actual\ A}{Limit\ A} + \frac{Actual\ B}{Limit\ B} + \frac{Actual\ C}{Limit\ C}\right)\right] * Limit\ A$$

$$Remaining\ B = \left[1 - \left(\frac{Actual\ A}{Limit\ A} + \frac{Actual\ B}{Limit\ B} + \frac{Actual\ C}{Limit\ C}\right)\right] * Limit\ B$$

$$Remaining\ C = \left[1 - \left(\frac{Actual\ A}{Limit\ A} + \frac{Actual\ B}{Limit\ B} + \frac{Actual\ C}{Limit\ C}\right)\right] * Limit\ C$$

Şekil 3.5 Kalan Ömür Hesabı

Kalan Ömür Hesabı:

1. Yöntem:

$$R = L - FC \tag{3.1}$$

R - Kalan ömür.

L - Belirlenen ömür sınırı.

FC - Uçtuğu uçuş sayısı/yaptığı devir sayısı.

2. Yöntem:

$$R = \left(1 - \left(\frac{\left(\text{FC} - \text{SUM(AFC)}\right)}{\text{SUM(L)}} + \text{P}\right)\right) * \text{L} \tag{3.2}$$

R - Kalan ömür.

L - Belirlenen ömür sınırı.

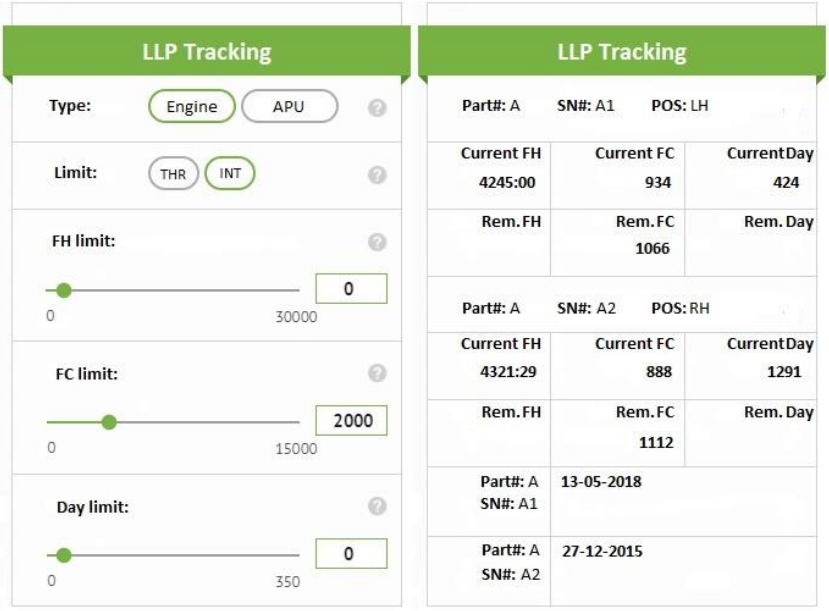
FC - Uçtuğu uçuş sayısı/yaptığı devir sayısı.

AFC - belirlenen sınırdan motorun uçtuğu uçuş sayısı.

P - Modül oran katsayısı.

Ömürlü parçaları bu bakım kartı altında üretildiği tarihten veya son bakım tarihinden itibaren saymaktadırlar. Kalan ömür değeri (remaining cycle) denklem 3.1'deki şekle göre hesaplanır. Her parçaya sınır (threshold/interval) değeri

belirlenir ve bu değerden parçanın uçtuğu toplam devir sayısı çıkartılarak kalan ömür bulunur. Sisteme uçuş eklendikçe devir sayısının artması ve kalan ömrün azalması sağlanmıştır. Kalan ömür toplam sınır değerinin %15-%20 altına düştüğünde tahmini bakım günü hesaplanarak motor bakım/söküm planı oluşturulur.



Şekil 3.6 Kalan Ömür Takibi

Bakım kartının altında sınırlı ömürlü parçaların raporu Şekil 3.7'de gösterilmiştir.

LIFE LIMITED REPORT

Part Name	Part Number	Serial Number	TSN	CSN	Life Cycle	Remain Cycles
IP REAR STUB SHAFT	UL20016	RRDSFTDAVR1582	23637:32	5074	26500	21426
IPC SHAFT	UL30176	DN65702	70451:32	19956	26100	6144
IPT DISC	UL27940	ADREB330	70451:32	19956	26500	6544
IPT SHAFT	UL37571	BAEF5654	70451:32	19956	26500	6544
LPC DISK	UL21343	STDK20188/A	70451:32	19956	22600	2644
LPC SHAFT	UL27910	PATH2908/A	70451:32	19956	27250	7294
LPT DISC STAGE 1	UL29028	EETM5746	70451:32	19956	27650	7694
LPT DISC STAGE 2	FW16689	SETM09775	2889:12	21774	26300	4526
LPT DISC STAGE 3	UL29031	SETY02984	70451:32	19956	25630	5674
LPT SHAFT	FW30433	BAEF5604/A	70451:32	19956	25200	5244

Şekil 3.7 Rapor Örneği

3.2.6 Motorun İtme Kuvveti

Motorun üreteceği en yüksek itiş gücün göstergesidir (uçağın yerden kalkması için gereken itiş gücü olarak ta tanımlanabilir). CFM-80E1A-X ailesinin itkisi 64530-

68530 lbf arasında değişiklik göstermektedir. Her motorun itiş gücün değeri üreticisi tarafından belirlenmektedir.

3.2.7 Mekanik Hasarlar

Hava şartlarından dolayı oluşan hasarlar (rüzgârla giren toz ve taş parçacıkları, dolu yağışı), korozyon, kuş çarpmaları, bakım sırasında teknisyenlerin motor içinde unutulan bakım malzemeleri motorlarda ağır hasarlara yol açabilmektedir.

3.3 Motor Bakımı

Motor Bakım, motorun kullanım ömrü boyunca güvenilir, uygun maliyetle çalışmasını sağlamak için gereken işi tanımlar. Uçak bakım maliyetleri, bir havayolu şirketinin işletme giderlerinin yaklaşık %10 - %15'ini temsil etmektedir, bunun %35 - %40'ı motorla ilgilidir. Malzeme değişimi, motor bakımındaki en önemli öğedir ve genellikle motorun doğrudan bakım maliyetinin %60 - %70'ini oluşturur. Bunun nedeni parçaların yıpranması, değiştirilmesi veya onarılmasıdır.

Motor bakımı üç ana nedenden dolayı gereklidir:

1. Operasyonel: Motoru gelir elde edebilmek için uygun ve güvenilir bir durumda tutmak.
2. Değer Tutma: Ömrü boyunca fiziksel bozulmalarını en aza indirerek motorun mevcut ve gelecekteki değerini korumak.
3. Düzenleme Gereklilikleri: En azından düzenleyici otoriteler tarafından istenen asgari denetim ve bakım standartlarını karşılamak.

3.4 Maliyet

Maliyet, özellikle rekabet gücü yüksek bir sektörde firmaların birincil faktörüdür. Havayolu operasyonları, ürün veya hizmetlerin maliyetlerini ve satın alma fiyatlarını doğrudan etkiler. Bu maliyetler doğrudan veya dolaylı olarak tedarik zincirinden elde edilir. Sonuç olarak, daha yüksek maliyetler ve fiyatlar havayolu rekabetçiliğini azaltır. Choy 2007 yılında, uçak parçalarının maliyetlerini araştırıp yeniden işleme, bakım veya değiştirme için çeşitli parçaların kullanımında lojistik akışının etkinliğini izlemek ve sınıfının en iyisi uygulaması ile kıyaslama yapmak için bir performans ölçüm sistemi geliştirmiştir [67].

Bununla birlikte, motor bakımı, havayollarının nakit akışındaki en önemli konudur. Havayolları motor bakım maliyetini en aza indirgemek için sürekli optimizasyon üzerinde çalışır. Maliyeti etkileyen faktörler aşağıda sıralanmıştır:

- Zaman; bakım süresi ve malzeme tedarik süresini doğru bir şekilde planlamak gerekir. Eğer bakım süresi yanlış planlanırsa uçak operasyona alınamaz, aynı şekilde eğer bakımda kullanılacak malzeme havayolu hangarlarında yok ise satın alınması ve bakım başlamadan önce hazır olması gerekir, eğer satın alma süreci uzarsa veya parça geç gelirse şirkete zarar verir.
- Kalite; uçak parçaları, sivil havacılık teşkilatı tarafından resmi olarak onaylanan fabrikalar tarafından üretilmelidir. Ayrıca, uçak parçalarının muayenesi, onarımı, değiştirilmesi veya elden geçirilmesi onaylı bir fabrika tarafından yapılmalıdır. Dünya çapında akredite edilmiş denetçiler Federal Havacılık İdaresi (FAA) ve Avrupa Havacılık Güvenliği Ajansıdır (EASA). Öte yandan, hava taşıtı bakımının kalitesi onay kuruluşu ile ilgilidir.
- Güvenilirlik; bir sistemin belirtilen koşullarda belirli bir süre için amaçlanan işlevini yerine getirme olasılığı olarak tanımlanır. Hava taşıtı güvenilirliğini artırmak için, bakım yöneticileri, hava taşıtı parçası tedarik süresi ve onarım zamanı ile ilgili olan bakım nedeniyle hava taşıtlarının durma süresini azaltmalıdır. Düşük güvenilirlik, daha sık düzeltici bakımı tetikleyen sık arızaları gösterir.
- Kullanılabilirlik; havayolları gelirlerini en üst düzeye çıkarmak için yüksek kullanım elde etme çabasındadır. En yüksek düzeyde kullanılabilirlikle sonuçlanan uçak bakım sürelerini minimize etmek için olası çözümler ararlar. Daha yüksek uçuş saatleri (daha düşük yer zamanı) daha yüksek kullanılabilirlikle sonuçlanır.
- Esneklik; üretim tesisi veya hizmet sağlayıcısının müşteri beklentisini karşılamak için planlanan işlemi başka bir sürece veya çözüme geçirme kabiliyetidir.

3.5 Optimizasyon Kuralları

Optimizasyon problemleri genellikle ifade etmesi kolay fakat çözümü zor olan problemler olarak tanımlanır. Matematiksel anlamı ise, amaç fonksiyonunu belirli kısıtlar altında minimum veya maksimum yapmaktır. Karar almak veya alternatifler arasından seçim yapmak insanların olduğu gibi işletmelerin de karşı karşıya kaldıkları bir durumdur. Genel olarak optimizasyon, verilen şartlar altında, en iyi sonucun elde edilmesi işidir. Son zamanlarda, optimizasyon problemlerini bir çözüm metodu olarak Yapay Sinir Ağları, Tavlama benzetimi algoritması, Tabu Arama ve Genetik Algoritmalar vb. yöntemler giderek artan bir ilgi görür.

3.5.1 Optimizasyon Kuralların Belirlenmesi

Bir motor üç ana sebepten ötürü uçaktan sökülerek (off-wing) atölye ziyareti (shop visit) şeklinde adlandırılan ağır bakım görür:

- Motor ömürlü parçaların ömürlerin bitmesi (LLP Expiry): Motor üzerinde takılı olan ömürlü parçalardan kalan ömrü en az olan LLP'nin (sınırlayıcı LLP) ömrünün dolacak olmasıdır. Her bir parçanın ömür değeri vardır, Model üzerinden sayan motorlarda 'Cycle Limit' olarak adlandırılır. LLP Task üzerinden sayan LLP parçalar da Threshold/Interval sınırı (limiti) olarak adlandırılır.
- Performans kaybı (Performance Loss): Motorun LLP'lerinin ömrü olsa bile, her motor tipinin kanat üzerinde operasyonuna devam edebileceği ortalama bir limit bulunur. Bu değerler (TSO/CSO) önceden tanımlıdır ve motor bu operasyon değerine gelince tamire gidecekmiş gibi hesap yapılır.
- Kiralama iadesi (Lease Return): Havayolunda operasyon yapan uçakların bir kısmı havayoluna aitken, büyük bir kısmı dış firmalardan kiralanmış olan uçaklardır. Her kira sözleşmesinde motorların geri teslim koşulları bulunur. Eğer ki motorlar kira süresinin sonunda geri teslim koşullarını sağlamazsa, tamire gitmek zorundadır.

Aşağıdaki kriterler motor söküm planında kullanılır:

- Günlük kullanım (Son 1 yıllık ortalama).
- Hiç ağır bakım görmemiş (1st run) motorlar için TSN/CSN bilgisi.

- Daha önce ağır bakım görmüş (2nd run ve 3rd run) motorlar için TSO/CSO bilgisi.
- Güncel LLP ömür sınırı (sınırlayıcı).

Aşağıdaki bilgiler her motor için hesaplanır:

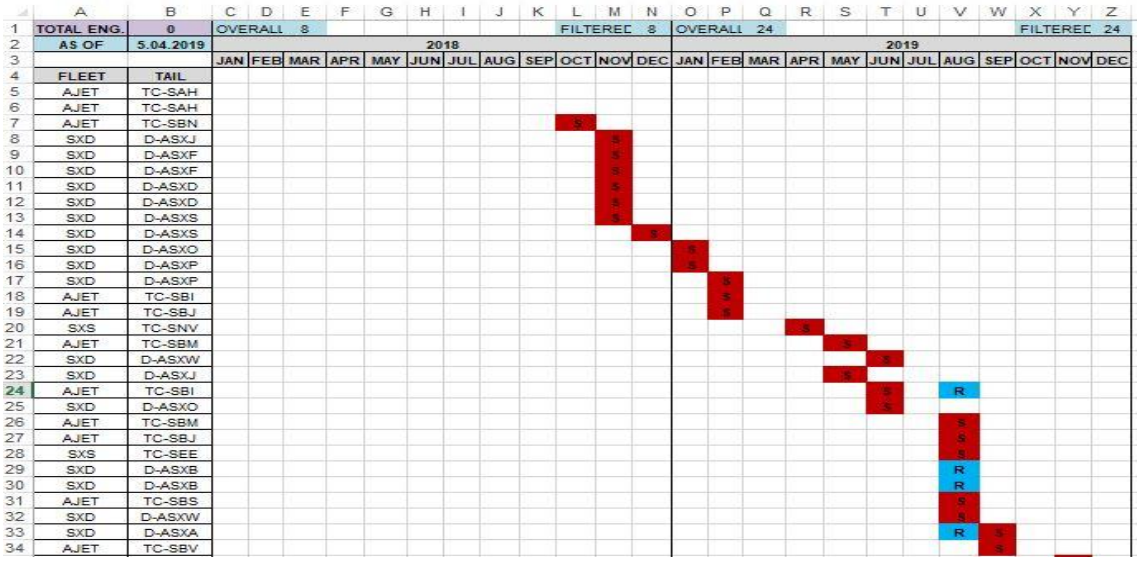
- LLP ömür sona erme tarihi.
- Performance Loss sonlanma tarihi.
- Kiralama bitiş tarihi ve geri teslim koşullarına uyup uymadığı.

Yukarıdaki üç kriter gözden geçirilip, ilgili motorun ağır bakım gerekip gerektirmediği, eğer gerecekse hangi kriterden dolayı ne zaman gerektirdiği hesaplanır. Bu kritere göre motor söküm planı elde edilir.

Motor söküm planı oluşturmak kolay bir süreç değildir. Birden çok kriter, sınır ve sonuçları analiz edilmesi gerekir. Uygulamanın en önemli ve kritik bölümüdür. Çünkü bu aşamaya kadar yapılan işlemler, sınırları belli olan, 4 işlemle çözümlenebilecek bir süreçten ibarettir. Basamaklandırma (Staggering) diye adlandırılan ve "STAGGER" olarak belirtilen Şekil 3.8'de gösterildiği gibi bu işlemi şirketler Microsoft Excel'de hesaplayarak tek tek tüm motorlar için yapmaktadır.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	TOTAL ENG.	126																
2	AS OF	5.04.2019																
3																		
4	FLEET	TAIL	POS	ESN	C	MRO	WS	SV_INI	REASON	SV_DUE	STAGGER	SPARE	SV-ST	H-L	P-L	P_TSN	R_CON	EGT
5	AJET	TC-SAH	2	894415	1	TEC		SV	SV	SV			SV	N/A	SV	SV	SV	SV
6	AJET	TC-SAH	1	894414	1	TEC		SV	SV	SV			SV	N/A	SV	SV	SV	SV
7	AJET	TC-SBN	2	894606	1	TEC		SV	SV	SV	25-Oct-18	TK_SPARE_5	SV	N/A	SV	SV	SV	SV
8	SXD	D-ASXJ	2	889309	1	TEC	LPC+LPT	SV	SV	SV	9-Nov-18	AJW_SXD_1	SV	N/A	SV	SV	SV	SV
9	SXD	D-ASXF	1	890701	1	KLM	LPC+LPT	SV	SV	SV	4-Nov-18	TO BE GROUNDED	SV	N/A	SV	SV	SV	SV
10	SXD	D-ASXF	2	890702	1	KLM	LPC+LPT	SV	SV	SV	6-Nov-18	TO BE GROUNDED	SV	N/A	SV	SV	SV	SV
11	SXD	D-ASXD	1	890754	1	KLM	LPC+LPT	SV	SV	SV	17-Nov-18	TO BE GROUNDED	SV	N/A	SV	SV	SV	SV
12	SXD	D-ASXD	2	890696	1	KLM	LPC+LPT	SV	SV	SV	20-Nov-18	TO BE GROUNDED	SV	N/A	SV	SV	SV	SV
13	SXD	D-ASXS	1	890755	1	KLM	LPC+LPT	SV	SV	SV	27-Nov-18	TO BE GROUNDED	SV	N/A	SV	SV	SV	SV
14	SXD	D-ASXS	2	890744	1	KLM	LPC+LPT	SV	SV	SV	4-Dec-18	TO BE GROUNDED	SV	N/A	SV	SV	SV	SV
15	SXD	D-ASXO	2	896309	1	TEC	LPC Damages	SV	SV	SV	15-Jan-19	WLFC_SXD_1	SV	N/A	SV	SV	SV	SV
16	SXD	D-ASXP	2	896319	1	KLM	Core+ LPT	SV	SV	SV	28-Jan-19	SXS_SPARE_1	SV	N/A	SV	SV	SV	SV
17	SXD	D-ASXP	1	897292	1	KLM	Core+ LPT	SV	SV	SV	1-Feb-19	WLFC_SXD_2	SV	N/A	SV	SV	SV	SV
18	AJET	TC-SBI	1	890324	1	TEC		SV	SV	SV	4-Feb-19	TK_SPARE_2	SV	N/A	SV	SV	SV	SV
19	AJET	TC-SBJ	1	891273	1	TEC		SV	SV	SV	8-Feb-19	TK_SPARE_4	SV	N/A	SV	SV	SV	SV
20	SXS	TC-SNV	2	874439	1	KLM	HPT Blade + TRF	SV	SV	SV	11-Apr-19	SXS_SPARE_2	SV	N/A	SV	SV	SV	SV
21	AJET	TC-SBM	2	890342	1	TEC		SV	SV	SV	8-May-19	TK_WLFC_1	SV	N/A	SV	SV	SV	SV
22	SXD	D-ASXW	1	896517	1	KLM	Core	31-Mar-19	PERF.	31-Mar-19	12-Jun-19	WLFC_SXD_2	-73	YOK	-161	OK	WARNING	YOK
23	SXD	D-ASXJ	1	888321	1	KLM	LPC+LPT	YOK	YOK	YOK	26-May-19	SXS_SPARE_1	#DEĞER!	YOK	YOK	YOK	YOK	YOK
24	AJET	TC-SBI	2	890263	1	TEC		31-Jan-20	RETURN	14-Aug-19	14-Jun-19	TK_SPARE_5	61	N/A	RETURN	OK	WARNING	N/A
25	SXD	D-ASXO	1	896308	1	KLM	Core+ LPT	1-Jan-20	PERF.	1-Jan-20	22-Jun-19	AJW_SXD_1	194	N/A	-361	OK	WARNING	OK
26	AJET	TC-SBM	1	891289	1	TEC		24-Aug-19	PERF.	24-Aug-19	2-Aug-19	TK_SPARE_2	23	N/A	-18	OK	WARNING	N/A
27	AJET	TC-SBJ	2	888380	1	TEC		6-Jul-19	PERF.	6-Jul-19	16-Aug-19	TK_SPARE_4	-40	N/A	-105	OK	WARNING	N/A
28	SXS	TC-SEE	2	888959	1	KLM	Fan + Core	21-Aug-19	LLP	21-Aug-19	14-Aug-19	SXS_SPARE_2	8	N/A	LLP	OK	WARNING	N/A
29	SXD	D-ASXB	1	874702	1	KLM	Core	21-Aug-21	RETURN	13-Aug-19	20-Aug-19	SXS_SPARE_3	-7	N/A	RETURN	OK	WARNING	OK
30	SXD	D-ASXB	2	892323	1	KLM	Core	17-Mar-22	RETURN	13-Aug-19	23-Aug-19	WLFC_SXD_1	-10	N/A	RETURN	OK	WARNING	OK
31	AJET	TC-SBS	2	891316	1	TEC		13-Jun-17	PERF.	13-Jun-17	23-Aug-19	TK_WLFC_1	-800	N/A	-844	OK	WARNING	N/A
32	SXD	D-ASXW	2	896520	1	KLM	Core	31-Mar-19	PERF.	31-Mar-19	27-Aug-19	SXS_SPARE_1	-149	YOK	-161	OK	WARNING	YOK
33	SXD	D-ASXA	1	874253	1	KLM	Off Wing TRF	27-Jul-22	RETURN	13-Aug-19	2-Sep-19	OFF WING TRF	-20	N/A	RETURN	OK	WARNING	OK
34	AJET	TC-SBV	1	888894	1	TEC		16-Oct-19	LLP	16-Oct-19	27-Sep-19	TK_SPARE_5	20	N/A	LLP	OK	WARNING	N/A

Şekil 3.8 Basamaklandırma, Motor Bakım/Söküm Hesabı



Şekil 3.9 Basamaklandırma

Basamaklandırma kavramına örnek verecek olursak, 2019-Nisan ayında bakım görmesi gereken 6 motor varsa, hepsini bir anda Nisan ayında yapmak yerine, bunların sökümü planlanır. Örneğin; 2 motor 2019-Ocak ayında, geriye kalan 4 motor 2019-Nisan'da yapılacak gibi planlanır. Şirketlerin elinde sonsuz yedek motor olmadığından ve motor pahalı bir yatırım olduğundan, basamaklandırma kavramına ihtiyaç duyulur.

Basamaklandırma adımında yapılan işlemler şunlardır:

- Motor bakım planına göre motor söküm planı çıkarılır.
- Motor söküm planında amaç, en ucuz maliyetle bir söküm planı gerçekleştirilmesidir.
- Bir motor bakım amaçlı söküldüğünde, uçağın operasyonu devam edebilmesi için yedek motor gereklidir. Çünkü motorun ağır bakımı yaklaşık 90-100 gün sürmektedir. Bazı durumlarda, özellikle düşük sezonda uçağı operasyondan çekip 100 gün yatırma opsiyonu da gerçekleşebilir. Bu durumda yedek motor ihtiyacı ortadan kalkar.
- Şirketler yedek motoru kendi envanterine satın alabileceği gibi kiralama (leasing) şirketlerinden de kiralayabilir. Örneğin, CFM56-7B tipindeki motorun satın alma maliyeti 12.000.000 USD iken, günlük kirası Pazar durumuna göre 2.000-2.500 USD arasında değişebilir.

- 2019-Nisan'da sökülmesi gereken motor, 2019-Ocak'ta sökülürse, bu motorların daha az kullanılması anlamına gelir ve finansal anlamda negatif sonuçtur. Ancak, bu şekilde şirket, daha az yedek motora ihtiyaç duyacağından, finansal anlamda pozitif bir durumdur.

Hedefimiz: Motor bakım/söküm tarihini bulduktan sonra, ilgili tarihi öne çekmek veya ötelemek mi, ya da sökülen motor yerine yeni motor satın almak mı, yoksa kiralamak mı en iyi sonucu sağlar sorusuna optimum yanıtı verebilmektir.

Talep kolay gözükse de, uygulamak göründüğü kadar kolay değildir. Filoda bulunan motorlarda ağır hasar veya operasyonel sorun yok ise basamaklandırma yapmamız pek zor olmayacaktır. Havayolu şirketin filo yönetiminde basamaklandırma uygulayabilmesi için, aşağıdaki faktörlerin incelemesi gerekir:

- Filoda uçuştaki motor sayısı;
- Yedek motor sayısı;
- Atölye ziyareti (Shop Visit) için bakım bütçesi;
- Hat bakımının iş yükü;
- Atölye durumu (Shop slot availability) ve TAT (turn-around time);
- Devir (Cycle) olarak motorun kanat üzerinde uçma süresi;
- Çift motorlu uçaklar için: iki motor arasındaki devir farkı.

Listenin tamamını oluşturmamakla birlikte, bu faktörler, her motor için sökme ve atölye ziyaretini planlamada kullanılan kritik bilgilerdir.

Bu bilgilere dayanarak bakım operatörü, hangi motorun uçabileceğine karar verir ve uçuş operasyonunu etkilemeyecek şekilde bakım planı hazırlar.

Örneğin: Bir operatör/havayolu 24 aylık sürede 16 uçak satın almış olsun. Uçakların üzerinde 32 motora ilave olarak üç yedek motor satın alsın. Operatör ayda ortalama 180 devir/cycle, yılda 2.100 devir yapsın, yani her motor yılda 2.100 devir biriktirsin. Örneğimize göre, motorların ortalama 10.000 devir sonra bir atölye ziyareti yapacağını varsayalım. Bu, beşinci veya altıncı yılda, tüm motorların atölye ziyareti için sökülmesi anlamına gelir. Toplam 32 motorun veya ayda yaklaşık iki motorun atölye ziyaretine göndermek demektir. Atölye ziyaretinin 60 gün (iki ay) dönüş süresi varsa, bu atölye ziyaretleri başladığında

operatörün iki motorunun hizmet dışı kalacağı anlamına gelir. Atölye ziyaretiyle ilgili herhangi bir sorun yoksa üç yedek motor fazlasıyla yeterli olacaktır.

Tablo 3.1 Motor Değişim Senaryosu 1

Ay	1	2	3	4	5	6	7	8
Servis dışı	1&2	3&4	5&6	7&8	9&10	11&12	13&14	15&16
Servis başlangıcı		Y1&Y2	1&2	3&4	5&6	7&8	9&10	11&12

Yukarıda ki tabloya göre 2. ayda 2 tane yedek motor takarak ilgili uçağın uçmasını sağlayabiliriz. Böylece bakıma tek bir uçağımız alınır ve 4 tane 13, 14, 15, 16 motorlarımız aynı anda atölyeye gönderilir.

Uzun tamir nedeniyle atölyede kalan motor olursa operatör ne yapar sorusu sorulduğunda veriler aşağıdaki gibi değişir:

Tablo 3.2 Motor Değişim Senaryosu 2

Ay	1	2	3	4	5	6	7	8
Servis dışı	1&2	3&4	5	7&8	6&9	11&12	13&14	15&16
Servis başlangıcı			1	2&3	4&8	6&7	5&9	11&12

Yukarıda ki tabloya göre 2 numaralı motorun servisten geri gelmesi 30 gün uzayınca, motorların takım planı otomatik olarak değişebilir. Bunun etkisi, uçakları havada tutmak için motorların uçaklar arasında hareket ettirilmesine neden olur. Havayolu şirketleri motorları uzun süreli tamire göndermek istemeyebilir, bu işlem havayolu için çok masraflıdır, motorları uçaklar arasında değiştirmeyi tercih edebilir.

Örneğin: iki tane uçağımız olsun, her bir uçağın bir motorun atölye ziyaret zamanını yaklaştığını düşünelim, eğer her iki uçağı da hangara göndererek motorların gelmesini beklersek 2 uçağımızda yaklaşık 2 ay uçamayacak demektir. Bu istenmeyen bir durumdur, uçağın yerde yatması havayolu için büyük bir zarardır, benzer durumlarda, ağır bakımı yaklaşan uçak seçilir ve bu uçaktan sağlam motor sökülerek diğer uçağa takılarak, uçağın operasyona geri dönmesi sağlanır.

İlk 12 ayda 24 tane motor atölye ziyareti yapmış olacak, her bir atölye ziyareti 2 milyon$, toplamda 48 milyon$. Bu durumu aşmak için iyi bir filo yönetim planı şarttır. Burada en önemli husus, filo yönetimi yaparken motor durum değerlendirmesi yaparak plan yapmaktır. Verilen örnekte daha önce bahsettiğim gibi, operatör ortalama olarak her 10.000 devir de bir atölye ziyareti yapıyor. Bu ortalama bir değer, bu yüzden bakıma daha erken gelecek motorlar da var ve bazıları daha sonra gelecektir. Doğru bir bakım planı hazırlandığında, operatör yıllık maliyet etkisini azaltabilir ve motorları daha uzun süre uçurabilir.

3.6 Optimizasyon Parametreleri

Motor bakım planlamasında, hangi kartların kullanılacağı, ömrü biten hangi LLP parçaların sökületeceği veya Overhaul'a gönderileceği, hangi uçakların motorlarına öncelik verileceği, hangi kaynakların planlamada dikkate alınacağı, optimizasyon çalışma sürecinde belirlenmesi gerekir. KDS skor hesabı yaparken eksik girilen veriye göre hesap yapar, veri güncellendikçe bulunan optimal değer tekrar hesaplanarak kullanıcıya o anki veriye göre doğru sonucu gösterir.

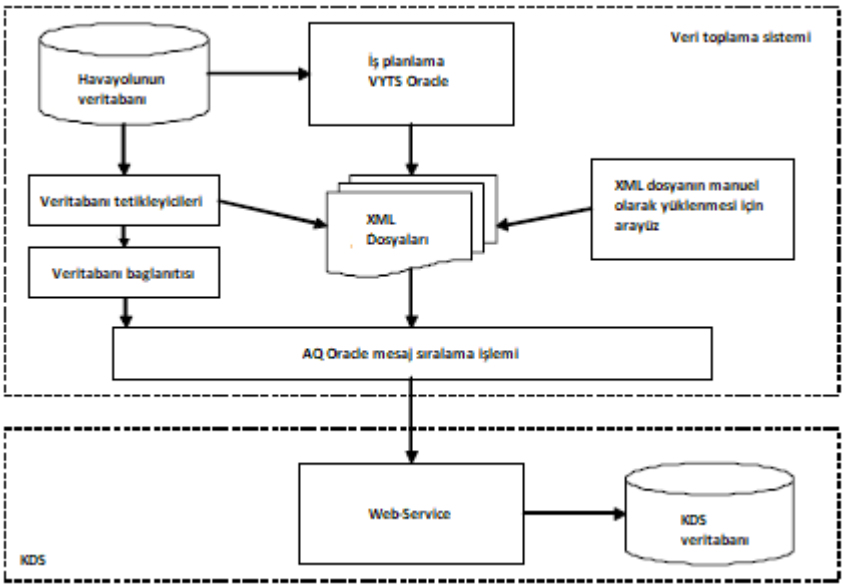
3.6.1 Veri Toplama ve Normalizasyon

Uygulamanın önemli bir girdisi olan gereksinim ve kaynak veri tabanın verilerin toplanmasını sağlayan veri toplama modülüdür.

Uygulamanın diğer sistemlerle de entegrasyonu sağlanarak ürüne daha geniş bir kullanım alanı sunulur. Diğer kurumsal kaynak planlama sistemleri ile kolay entegrasyon sağlanır.

Aşağıdaki entegrasyon seçenekleri görülmektedir:

- Veritabanı bağlantısı sağlanır ve veritabanı işlemleri yapılır.
- XML ve Excel ile veri içe/dışa aktarma.



Şekil 3.10 KDS'nin Havayolu Veri Tabanıyla Entegrasyon Şeması

Oluşturduğumuz yapının amacı sadece motor bakım/söküm zamanlarının hesaplanması değil, çok etkin bir kısıtlı ömür parça takip sistemi ve doğru zamanda motor temin etme süreç yönetimi sağlamaktır. Böylece çok çeşitli veri kaynaklarından ham verilerin derlenerek KDS veri tabanına aktarımı sağlayıp sonuçları belirleyebiliriz.

Avantajları:

- Şirketlerin motor satın alma veya kiralamada maliyet düşürülmesi.
- Etkin satın alma kararları.
- Etkin motor bakım planlama süreci.
- "Olursa ne olur?" metodu (what-if).
- Arıza (defect) analizi.

3.6.2 Optimizasyon Yöntemlerin Analizi

YSA öğrenme süreci, nöronlar arasındaki bağlantı ağırlıklarını ayarlayarak gerçekleştirir. YSA öğrenmenin amacı, eğitim ve test örnekleri üzerindeki nesnel işlevi en aza indiren ağırlıklarının durumunu bulmaktır. YSA öğrenme süreci oldukça karmaşıktır ve bunun nedeni aşağıda sıralanan maddelerden kaynaklanır:

- Bir YSA çok sayıda bağlantı içerebilir; bu yüzden sinir ağının muhtemel durumlarının alanı çok büyüktür ve hepsini düşünmek neredeyse imkânsızdır.
- Gizli bir katmanda birden fazla nöron içeren sinir ağları, bir değil birkaç optimum koşula sahiptir. Bu koşulların bazıları çok az hata değeri içerir, ancak, bir tanesi gerçekten istenilen minimum hata değerine ulaştığında optimum durum olarak kabul edilir.
- Çok sayıda yerel minimum noktaların, küresel bir optimum arayışını önemli ölçüde zorlaştırır.

Günümüzde, sinir ağlarının parametrelerini optimize etme problemini çözmek için çok sayıda farklı yöntem geliştirilmiştir. Temel olarak, bu yöntemler birinci dereceden yöntemlerdir ve Gradyan İniş yöntemin çeşitleridir. Optimizasyon yöntemlerinden bir kaçını inceleyelim:

Gradyan İniş

Yöntemin ana fikri en hızlı iniş yönünde gitmektir (yerel minimum noktasını bulma) ve bu yön antigradient (gradyan inişine zıt olan ve dolayısıyla en hızlı düşüşüne doğru yönelmiş vektördür) tarafından ayarlanır. Ardından, i-yinelemedeki ağırlık değişiminin δw ifadesi aşağıdaki forma sahiptir:

$$\delta wn = -\lambda n \nabla wn-1 L(wn-1) \quad (3.3)$$

Bu formülde:

λn - n-inci yinelemede/iterasyonda öğrenme hızı katsayısıdır.

L - fonksiyonun gradyanıdır.

w – ağırlık.

$n-1$ - İterasyon sayısıdır.

Gradyan iniş yöntemi, nesnel L fonksiyonun yerel minimum noktasında takılabilir. Bunun üstesinden gelmek için, moment veya atalet yöntemi kullanılır, yenilik "atalet" terimini eklemekten ibarettir:

$$\delta wn = -\lambda n \nabla wn-1 L(wn-1) + \alpha \cdot \delta wn-1 \quad (3.4)$$

Bu formülde:

α – atalet katsayısı, sabit bir değer alabilir ve λn'ye bağlı olarak değişebilir, örneğin $\alpha n=1-\lambda n$;

δwn - 1 - önceki yinelemede ağırlık değişimi [68].

AdaGrad

Bu yöntem 2011 yılında ortaya atılmıştır [69]. AdaGrad'ın özü her parametrenin öğrenme oranı bu parametrenin gradyanlarının geçmişine göre ölçeklendirilmesinden oluşur. Bu ölçeklendirme, mevcut gradyanı önceki gradyanların karelerinin toplamına bölerek yapılır. AdaGrad yöntemini kullanarak ağırlık değiştirme formülü aşağıdaki gibidir:

$$Gn=Gn-1+(\nabla wn-1L(wn-1))2$$

$$\delta w_n=-\lambda_n\sqrt{G_n+\varepsilon}\cdot\nabla_{w_{n-1}}L(w_{n-1}) \quad (3.5)$$

Bu formülde:

Gn - gradyanların karelerinin toplamıdır;

ε - düzeltme katsayısı, $0<\varepsilon\ll1$.

Buradaki fikir, çok sık güncellediğimiz öğeler için güncellemeleri azaltan bir eleman kullanmaktır.

Adam

Adam yöntemi [70], AdaGrad'ın genelleştirilmesi olarak kabul edilebilir. Hem hareket biriktirme fikrini hem de tipik işaretler için daha zayıf bir ağırlık yenileme fikrini birleştirir. Adam yöntemi bir gradyanı aşağıdaki gibi dönüştürür:

$$mn=\beta1mn-1+(1-\beta1)\nabla wn-1L(wn-1)$$

$$vn=\beta2vn-1+(1-\beta2)\nabla wn-1L(wn-1)$$

$$m\hat{}n=mn1-\beta1n,\quad v\hat{}n=vn1-\beta2n$$

$$\delta w_n=-\lambda_n m_t\hat{}\sqrt{v_t\hat{}}+\varepsilon \quad (3.6)$$

Bu formülde:

mn - beklentinin tahmini;

vn –merkezli olmayan (noncentered) dağılımın tahmini;

β1 ve β2 biriktirme pencereleri, 0 < β1 <1 ve 0 < β2 <1;

$m\hat{}n$ ve $v\hat{}n$ - beklenti tahminlerinin düzeltilmiş değerleri ve merkezli olmayan gradyanın ortalaması, öğrenmenin ilk adımlarını etkiler;

ε - düzeltme katsayısı, 0 < ε≪1.

Adam yöntemi yazarları varsayılan değerleri olarak β1 = 0.9, β2 = 0.999, ε = 10^{-8} önermektedir ve algoritmanın, ilk kalibrasyon nedeniyle geniş bir veri setinde önceki tüm algoritmalar ile daha iyi veya yaklaşık aynı performans gösterdiğini iddia etmektedir [70].

Etkili bir yöntem seçimi, yakınsama oranı kriterine göre birkaç optimizasyon yönteminin test edilmesinin sonuçları temelinde yapılmıştır.

Öğrenme basamağı tüm yöntemler için 0,001 olarak belirlenmiştir. Ataletli gradyan inişi için atalet katsayısı, aşağıdaki formülle hesaplanmıştır: 1 - λ. Adam yöntemi için β1 ve β2 parametreleri sırasıyla 0,1 ve 0,001 değerlerine eşitlenmiştir.

Tablo 3.3 Optimizasyon Yöntemlerin Öğrenme Değerleri

Yöntem	Öğrenme	Test
Gradyan İniş	1,136	1,108
Ataletli gradyan inişi	0,140	0,130
AdaGrad	0,370	0,343
Adam	0,050	0,049

Elde edilen sonuçlara göre, en etkili optimizasyon yönteminin Adam yöntemi olduğu açıktır. Diğer yöntemlere kıyasla en yüksek yakınsama oranını göstermiştir.

3.7 Sistemin Tanıtılması

Motor performansını sıcaklık, itme, güç değerleri, hava sıkıştırma oranı ve yakıt miktarı belirler. Bu faktörler direk maliyeti etkilediğinden, motor bakım sürecin takibi havacılıkta önem arz etmektedir. Motor bakımı belirli aralıklarla planlı olarak takip edilmesi ve plansız tespit edilen arizaların üretici firma talimatları doğrultusunda onarılması gereklidir. Motorun bakımı (Overhaul) her 5 senede bir

kere yapıldığından ve maliyeti yaklaşık 3,5-4 milyon$ olduğunu hesaba katıldığında, motorun 5 sene içerisindeki bakım sayısı ne kadar azaltılabilirse, karlılık açısından o kadar verimli kullanılmış olur.

3.7.1 KDS Çözüm Algoritması

Algoritmanın kısa açıklaması:

- Verilerin tanımı ve girişi, veri analizi, kriterlerin ve önceliklerin belirlenmesi.
- Kriterlerin değerlendirilmesi, alternatiflerin belirlenmesi, karar vericinin tercihleri.
- Elde edilen göstergeler için arama limitlerinin belirlenmesi.
- P kavramının kullanımı, duruma göre 0 veya 1'dir
- Giriş sinyallerini seçilen örnek ile karşılaştırılır.
- Sinyallerin eşitliği durumunda, p_i=1, aksi halde p_i=0 anlamına gelir.
- Bir sonraki sinyal çiftine gidilir i+1.
- Son sinyale kadar kontrol edilir.
- Aşağıdaki formüle göre mevcut durumun seçilen örnek ile benzerlik derecesini hesaplanır:

$$S = \frac{\sum_{i=1}^{n} w_i z_i p_i}{\sum_{i=1}^{n} w_i' z_i'} \, 100\% \qquad (3.7)$$

S - mevcut durumun seçilen örnek ile benzerlik derecesidir, yüzde olarak ifade edilir;

w_i – mevcut durumun i-nci değerin ağırlık değeri;

z_i- mevcut durumun i-nci değerin gösterge değeri;

p_i- benzerlik durumu;

w_i' – i-nci örnek değerin ağırlık değeri;

z_i' – i-nci örnek değerin gösterge değeri.

3.7.2 Uygulama Modeli

Motor bakımında, motorların ve alt parçaların takibi kullanıcı tarafından belirlenen birden fazla kritere göre yapılır ve tüm kriterler, ölçülebilen veya

hesaplanabilen motor parametreleri ve ölçülemeyen kriterler olarak ikiye ayrılır. Bu çalışmada, ölçülebilen ve ölçülemeyen kriterlerin bir arada değerlendirilebilmesi ile motor bakım planı için YSA ve BM kullanarak KDS geliştirilmiştir.

Geliştirilen model, 82 uçağın verisini kullanarak, her uçakta iki motor olmak üzere, 164 motorun verileri içerisinden 18 farklı kritere göre motorların bakıma gönderme önceliğini bulmak için kullanılmıştır. Amacımız, önem arz eden kriterleri ağırlıklandırmak, kriterler ve kanıtlara dayanan alternatifleri derecelendirmektir. Kriterler alternatifler hakkında hiçbir bilgi olmadan ağırlıklandırılmalıdır. Değerlendiricilerin, tercih ettikleri alternatiflerin lehine kriter ağırlıklarının ağırlıklandırılmasını engellemektir. Kriterleri ağırlıklarına göre sıralayarak elde edilen sonuçların daha hassas olması elde edilmiştir. Çalışma sonucunda elde edilen sonuçlar, uzman kişi tarafından sağlanan ve diğer sistemlerin oluşturduğu sonuçlar ile karşılaştırılmıştır. Karşılaştırmalar sonucunda uzman kişinin sağladığı sonuçlar ve geliştirilen KDS'in bulduğu sonuçlar arasında fark %5 altında olduğu belirlenmiştir.

Geliştirilen KDS, on beş adet kritere göre yüz altmış dört adet motor için uygulanmıştır. Uygulama, Oracle PL/SQL kullanılarak gerçekleştirilmiştir. Kullanıcı arayüzü ile sorgulama ekranları HTML tabanlıdır, JavaScript ve JQuery teknolojileri kullanılmıştır. Raporlama motorlarından açık kaynak raporlama sistemi olan Business Intelligence and Reporting Tools rapor servisi kullanılarak gelen talepler, HTML, PDF, XLS, XLSX, DOC, DOCX formatlarında çıktı dokümanları elde edilir.

Belirlediğimiz kriterleri ölçülebilen ve ölçülemeyen kriterleri olarak ikiye ayırmıştık. Motor bakım/söküm planında değerleri ölçmeler yaparak veya hesaplayarak elde ettiğimiz kriterler şunlardır:

- Motor basınç oranı (MBO),
- N1 hız,
- EGS,
- LLP ömrü,
- Performance Loss (PL),
- Lease Return date (LR),

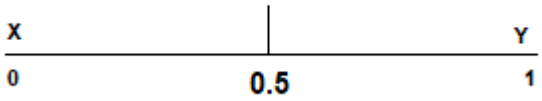
- Günlük Kullanım (GK),
- 1^{st} run,
- 2^{nd} run,
- 3^{rd} run.

Örnek verilirse: MBO, EGS motor başında teknisyenler tarafından ölçülerek; LLP ömrü, 1^{st} run ise hesaplanarak elde edilen değerlerdir. Her kritere üçgen üyelik fonksiyonları ve yamuk üyelik fonksiyonları kullanılmıştır**.**

Ölçülemeyen kriterleri ise:

- Arıza kaydı;
- Malzeme (materyal) kaybı;
- Boroskop sonucu (çatlak ve korozyon ölçüm değerleri);
- Rotor hızı;
- Yakıt tüketimi;
- Motor gücü (hava sıcak ise motor gücü ortamdaki sıcaklığa bağlı olarak azalır ve daha çok yakıt tüketir);
- Yabancı cisimden oluşan hasarlar (kuş çarpması, motora taş veya kum girmesinden oluşan deformeler, vb.);
- Hava şartlarından dolayı kalan ömür hesabının değişmesi.

Ölçülemeyen kriterler ölçülebilen kriterler gibi karar vermede önemli yer almaktadır. Örneğin, eğer 1 ay boyunca aynı motorda 3 tane benzer arıza kaydı tespit edilirse, motor uçaktan sökülebilir; boroskop ile yapılan ölçümler sırasında eğer bulunan çatlak veya korozyon ölçüm değeri belirlenen ortalama değerden daha fazla ise veya hava şartlarından dolayı materyal kaybı tespiti yapılırsa motor bakıma gönderilir. Motor performansını etkileyen kriterlerden birisi, harcanan yakıt olduğundan, yakıt tüketiminin de takibi gerekmektedir. Yakıt tüketimi, saatlik yakıt tüketiminin güce oranı ile belirlenir. Havayolları yakıt maliyetlerini düşürebilmek için çaba göstermektedirler. Örneğin, tüm kriterlere ilişkin ağırlıklar aşağıda açıklanan yöntemle belirlenir. Kriterler önem önceliğine göre karşılaştırılırken, hangi kriter diğerine göre daha önemliyse, o kriterin olduğu taraftaki bir derecelendirme görüşülen uzmanlarca “x” işareti ile işaretlenmiştir.



Şekil 3.11 Bulanık Eşleme Yaklaşımı

Y'ye göre X'nın tercih derecesi, R_{XY}, x işaretinden X'e olan uzaklıkla ölçülür. X'den Y'ye toplam uzaklık 1'dir.

Kriter ağırlıkların belirlenmesi:

- Eğer R_{XY} < 0,5 ise Y>X, μ üyelik derecesi 1'e yakındır.
- Eğer R_{XY} = 0,5 ise X≈Y, μ üyelik derecesi 0,5'e yakındır.
- Eğer R_{XY} > 0,5 ise X>Y, μ üyelik derecesi 0'a yakındır.

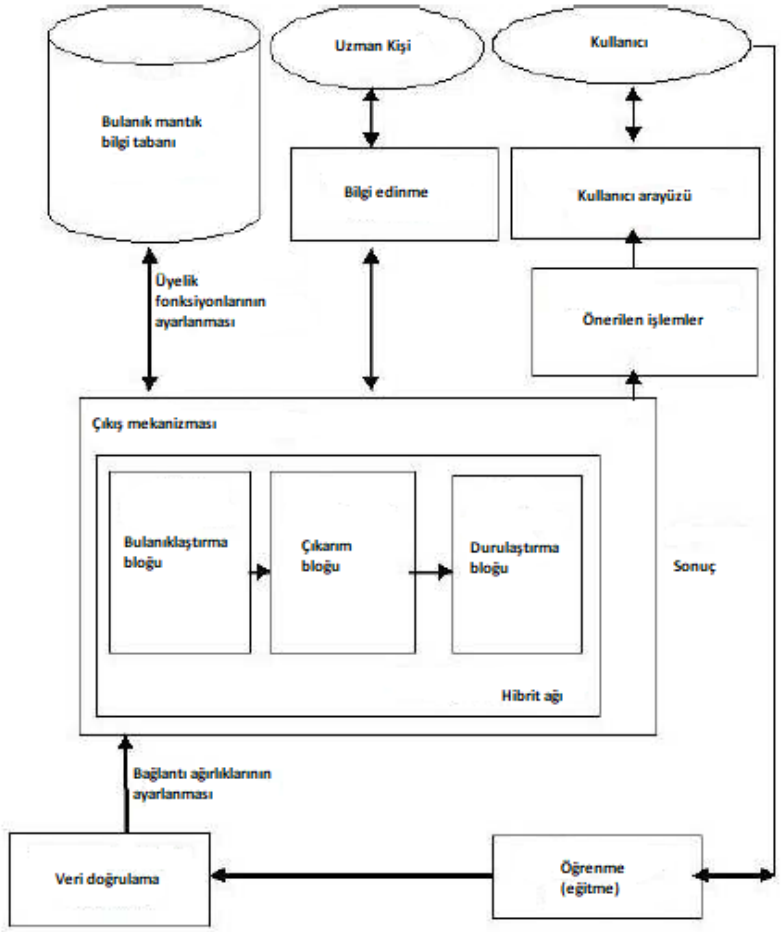
Kesin tercihler olması durumunda R_{XY} = 1 veya R_{XY} = 0.

Kriter ağırlıkları aşağıdaki formül kullanılarak bulunur:

$$\overline{\mu_1}=\frac{\mu_1}{\mu_1+\mu_2+\cdots+\mu_n} \tag{3.8}$$

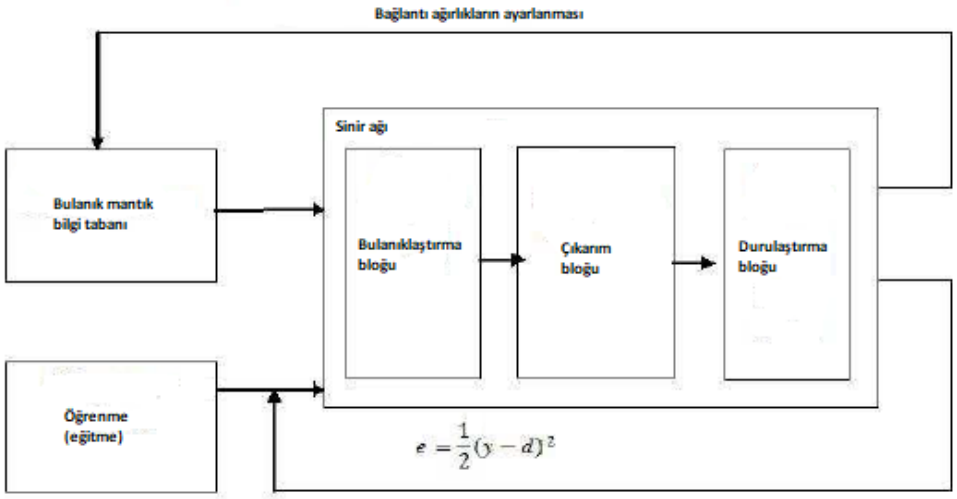
Ölçülebilen ve ölçülemeyen kriterler girdi olarak alınır, ardından bulanıklaştırılır ve YSA gönderilir. KDS tanımlanan kurallara ve sınırlara göre yapılan değerlendirmede çözümler bulunur.

Bulanık mantık ve sinir ağlarının birleşimi ile oluşturulan sistemler, tek bir yöntemle geliştirilen sistemlere göre avantajlıdır. A.Tortum ve arkadaşları, yaptıkları çalışmada her iki yöntemi birleştirerek olumlu sonuçlar elde etmişleridir [84].



Şekil 3.12 Nöro-bulanık Sistemin Yapısı

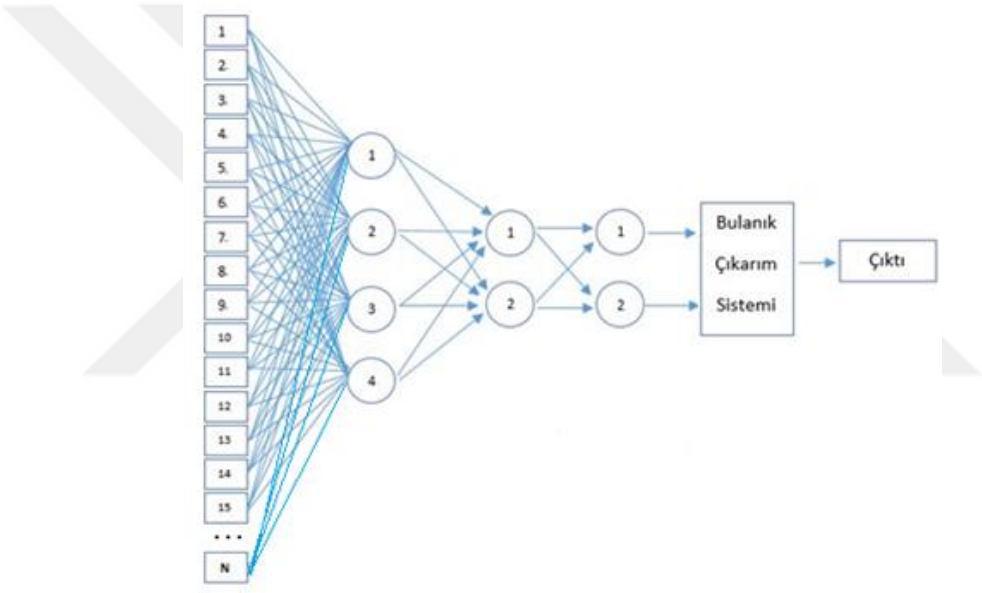
Şekil 3.13'te Üyelik fonksiyonunu düzeltmek için Nöro-bulanık hibrit ağ yapısı gösterilmektedir.



Şekil 3.13 Nöro-bulanık Hibrit Ağın Genel Yapısı

Yapılan işlemler aşağıda sıralanmıştır:

I. Verilerin toplanması.
II. Bulanıklaştırma. Kriterler üyelik fonksiyonları yardımı ile bulanıklaştırılmıştır.
III. Kriter ağırlıklarının hesaplanması.
IV. Bulanık değerlerin YSA değerlendirilmesi.
V. YSA'dan gelen sonuçların BÇS'de değerlendirme. Mamdani çıkarım tabanlı kurallara göre yorumlanır. Motor söküm planında, söküm/bakım tarihi beş değerle belirlenir: Çok yakın, yakın, normal, uzak, çok uzak olarak ifade edilir.



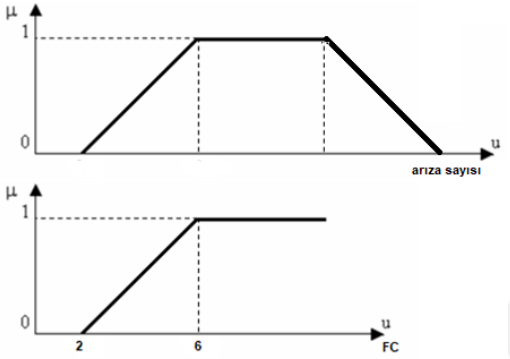
Şekil 3.14 YSA Modeli

Bulanık kümeyi belirlemek için farklı üyelik fonksiyonları kullanabilirsiniz, ancak pratikte sayısal hesaplamaları basitleştiren matematiksel işlevler biçiminde analitik bir gösterimi olanları kullanmak daha uygundur. Üyelik fonksiyonlarını tanımlamak için birden fazla eğri şekli vardır. En çok kullanılan parçalı doğrusal üyelik fonksiyonları (üçgen ve yamuk) ve Gauss üyelik fonksiyonları.

Yamuk üyelik fonksiyonu üç düz çizgiden oluşur ve dört parametre ile belirlenir (a,b,c,d). a ve d parametreleri, yamuğun alt tabanını, b ve c parametreleri ise yamuğun üst tabanını karakterize eder. Bu üyelik fonksiyonu, aralık (a, d), sınırlar

(a, b) (c, d) ve çekirdek [b, c] olan bir bulanık küme oluşturur. X noktasında ki değer aşağıdaki formüle göre hesaplanır:

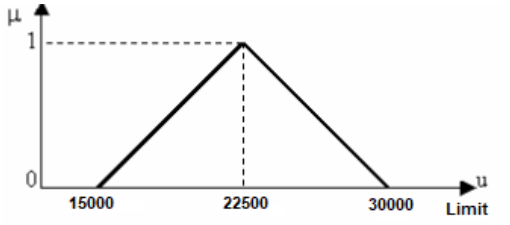
$$\mu(x) = \begin{cases} 1 - \frac{b-x}{b-a}, a \le x \le b, \\ 1, \quad b \le x \le c, \\ 1 - \frac{x-c}{d-c}, c \le x \le d, \\ 0, \text{diğer durumlarda} \end{cases} \tag{3.9}$$



Şekil 3.15 Üyelik Fonksiyonları

Üçgen üyelik fonksiyonu iki düz çizgiden oluşur ve üç parametre ile belirlenir (a, b ve c). a ve c parametreleri, üçgenin tabanını, b parametresi ise tepe noktasını karakterize eder. Bu üyelik fonksiyonu, aralık (a, c), sınırlar (a, c)\{b}, çekirdek {b} ve b modu olan bir bulanık küme oluşturur. X noktasında ki değer aşağıdaki formül 3.10'a göre hesaplanır:

$$\mu(x) = \begin{cases} 1 - \frac{b-x}{b-a}, a \le x \le b, \\ 1 - \frac{x-b}{c-b}, b \le x \le c, \\ 0, \text{diğer durumlarda} \end{cases} \tag{3.10}$$



Şekil 3.16 LLP Ömür Kriteri için Üyelik Fonksiyonu

Aşağıdaki Tablo 3.4'te ölçülebilen kriter değerleri sıralanmıştır:

Tablo 3.4 Motor Kriterlerin Ölçüm Değerleri

Kriter/ Motor	MBO	N1 hız %	EGS	LLP Devir	PL devir	LR	GK devir	1st run	2nd run	3rd run
A1	680	98.6	72	14000	13000	2.12.19	8	13999	5030	0
A2	685	100.4	68	13000	12000	2.12.19	8	11980	12986	0
A3	650	99.5	72	15000	14000	16.12.19	8	14999	14464	4000
A4	692	100.2	66	16000	15000	16.12.19	8	15695	2560	0
A5	674	100.8	50	17000	16000	16.12.19	8	16910	16960	500
A6	660	98.6	49	17000	16000	1.12.20	8	16995	16996	323
A7	685	101.4	49	17000	16000	1.12.20	8	16996	8943	0
A8	660	99.2	51	18000	17000	1.12.20	8	17990	13820	0
A9	685	98.6	48	18000	17000	1.12.20	8	14737	0	0
A10	689	100.8	46	18000	17000	1.12.20	8	17991	16290	0
A11	660	108.5	51	20000	19000	1.12.22	5	19029	0	0
A12	682	101.4	57	20000	19000	1.12.22	5	19999	3980	0
A13	688	99.2	60	20000	19000	1.12.22	5	20000	8392	0
A14	674	99.5	48	25000	24000	1.12.22	4	26060	2060	0
A15	670	103.5	58	25000	24000	1.12.22	4	34574	10574	0
A16	682	101.4	60	25000	24000	23.05.23	4	28203	4303	0
A17	689	101.5	47	25000	24000	23.05.23	4	23900	2000	0
A18	680	99.8	48	30000	29000	24.06.23	4	17492	0	0
A19	688	103.5	50	30000	29000	24.06.23	4	17492	0	0
A20	690	100.8	46	30000	29000	24.06.23	4	10760	0	0

Kriter ağırlıkların hesaplama sonucu aşağıdaki tabloda verilmiştir:

Tablo 3.5 Kriter Ağırlıkları

Kriter	Ağırlık
MBO	0,045
N1 Hız	0,032
EGS	0,046
LLP	0,098
PL	0,052
LR	0,065
GK	0.076
1st run	0.080
2nd run	0.074
3rd run	0,074
Ariza	0.098
Malzeme kaybı	0.047
Boroskop	0.105
Hasar	0.029
Hava şartları	0.014

Kriterleri matris ile ifade edersek, Tablo 3.6'daki gibi bir yapı söz konusu olabilir. Girdi1 fiziksel, girdi2 ise tahmin kriterlerini temsil etmektedir.

Tablo 3.6 Kriter Matrisi

	ÇY	***Y***	***N***	***U***	***ÇU***
ÇY	*ÇY*	*ÇY*	*D*	*N*	*N*
Y	*ÇY*	*ÇY*	*D*	*N*	*N*
N	*Y*	*Y*	*N*	*U*	*U*
U	*N*	*N*	*U*	*ÇU*	*ÇU*
ÇU	*N*	*N*	*U*	*ÇU*	*ÇU*

Not: ÇY-çok yakın, Y-yakın, N-normal, U-uzak, ÇU-çok uzak.

1) Eğer 1.veri girişi yakın ise ve 2. veri girişi yakın ise başarım çok yakındır.
2) Eğer 1.veri girişi yakın ise ve 2. veri girişi normal ise başarım yakındır.
3) Eğer 1.veri girişi yakın ise ve 2. veri girişi uzak ise başarım normaldir.
4) Eğer 1.veri girişi normal ise ve 2. veri girişi yakın ise başarım yakındır.
5) Eğer 1.veri girişi normal ise ve 2. veri girişi normal ise başarım normaldir.
6) Eğer 1.veri girişi normal ise ve 2. veri girişi uzak ise başarım uzaktır.
7) Eğer 1.veri girişi uzak ise ve 2. veri girişi yakın ise başarım normaldir.
8) Eğer 1.veri girişi uzak ise ve 2. veri girişi normal ise başarım uzaktır.
9) Eğer 1.veri girişi uzak ise ve 2. veri girişi uzak ise başarım çok uzaktır.

BÇS ile elde edilen sonuç değerleri Tablo 3.7'de verilmiştir. Bu değerler Tablo 3.4'de verilen kriterlere dayanarak oluşturulmuştur, eğer Tablo 3.4'de en az bir kriter değişirse, örneğin günlük kullanım 4'ten 8'e çıkarsa bakım/söküm günü 2 kat yaklaşır. Ek olarak bakım/söküm tarihlerini etkileyen ölçülemeyen kriterlerde erken bir tarihte motoru bakıma gönderme sebebi olabilir. Kurduğumuz yapının dinamik olması sayesinde KDS, arka planda güncellenen her veriye göre gerekli hesaplamaları yaparak kullanıcıya en güncel bakım/söküm tarihi bildirmekte, kullanıcıya tekrar işlem yaptırmamaktadır.

Tablo 3.7 KDS'nin Hesapladığı Bakım/Söküm Tarihleri

Motor	**Bakım/Söküm Tarihi**
A1	26-05-2022
A2	06-05-2019
A3	07-07-2019
A4	06-12-2023
A5	22-12-2024
A6	06-05-2019

A7	01-02-2022
A8	04-10-2020
A9	12-06-2020
A10	30-11-2019
A11	11-11-2019
A12	07-02-2028
A13	07-09-2025
A14	12-01-2035
A15	15-03-2029
A16	30-06-2033
A17	27-01-2035
A18	22-11-2027
A19	22-11-2027
A20	01-06-2032

Geliştirilen KDS ile aynı işlevi yapan, İsviçre kökenli havacılık yazılımı şirketi tarafından geliştirilen "Optimizer" sistemi ve Türk yazılımcıları tarafından geliştirilen "SV-Plan" (System Verification Plan) başarımları karşılaştırılmıştır.

Optimizer depth-first-search ve branch-and-bound algoritmaları kullanarak en optimal çözümü bulmayı hedefler. SV-Plan ise karar ağacı öğrenme algoritması üzerine kurulmuştur. Tasarladığımız KDS, SV-Plan'a göre daha hızlı ve daha net sonuçlar vermektedir. SV-Planda sapma oranı çok fazla olduğundan, net bir değer her zaman elde edilemez.

Optimizer ile karşılaştırıldığında, üretilen sonuçlar hemen hemen aynıdır. Optimizer hata oranı %7-8. Geliştirilen KDS daha net bir sonuç vermektedir. Sebebi, bakım/söküm tarihi hesabı yapılırken uçağın uçtuğu devir sayısına ek olarak uçuşların gün/saat/dakika verilerine bakılarak analiz edilmesidir. Kullanıcı tarafından eksik veri girildiğinde veya dış servislerden veri aktarımında sorun çıktığında, KDS eksik verinin ortalama değerini bularak planlama işlemine devam edebilmektedir. Mevcut sistemleri ile sonuçları karşılaştırdığımızda elde ettiğimiz sonuç verilerin optimal değerlere daha yakın olduğu tespit edilmiştir.

Tablo 3.8 KDS Sonuç Karşılaştırılması

Motor	**Optimizer**	**SV-Plan**	**KDS**
A1	20-05-2022	27-05-2022	26-05-2022
A2	03-05-2019	06-05-2019	06-05-2019
A3	04-07-2019	08-07-2019	07-07-2019
A4	05-12-2023	06-12-2023	06-12-2023
A5	22-12-2024	23-12-2024	22-12-2024
A6	05-05-2019	06-05-2019	06-05-2019
A7	04-02-2022	02-02-2022	01-02-2022
A8	03-10-2020	04-10-2020	04-10-2020

A9	10-06-2020	12-06-2020	12-06-2020
A10	27-11-2019	29-11-2019	30-11-2019
A11	10-11-2019	11-11-2019	11-11-2019
A12	09-02-2028	07-02-2028	07-02-2028
A13	06-09-2025	06-09-2025	07-09-2025
A14	12-01-2035	12-01-2035	12-01-2035
A15	14-03-2029	15-03-2029	15-03-2029
A16	29-06-2033	29-06-2033	30-06-2033
A17	28-01-2035	27-01-2035	27-01-2035
A18	23-11-2027	22-11-2027	22-11-2027
A19	23-11-2027	22-11-2027	22-11-2027
A20	03-06-2032	02-06-2032	01-06-2032

KDS'nin bulduğu bakım/söküm tarihine göre motorlar en yakın tarihten uzağa doğru sıralanarak Tablo 3.9'da verilmiştir.

Tablo 3.9 Sıralı Sonuç Değerleri

A2	1	A10	6	A5	11	A15	16
A6	2	9	7	A13	12	A20	17
A1	3	A8	8	A18	13	A16	18
A3	4	A7	9	A19	14	A14	19
A11	5	A4	10	A12	15	A17	20

3.7.3 Basamaklandırma İşlemin Sonucu

Doğru bir bakım/söküm planı oluşturmak için eksiksiz veri girişlerine ihtiyaç duyulur. Uçakların uçuş bilgileri servis sağlayıcıları tarafından gönderilmektedir, ilgili veri her 30 dakikada bir güncellenmektedir, eğer uçağın rotası değişirse, uçağın uçuş saati gecikmeden veya herhangi bir arızadan dolayı değişirse, uçuş farklı uçağa aktarılırsa veya uçuş komple iptal olursa, güncel duruma ihtiyacımız olduğundan, belirli aralıklar ile veritabanları arasında bağlantı oluşturarak güncel veri çekilir, eskisinin yerine yazılır.

Uçuşlarla birlikte, her uçağın yaklaşan planlı bakımları da listelenir. Eğer motor bakım/söküm tarihi planlanan bakım ile aynı tarihe veya ± 7 gün içerisindeyse, işlemleri birlikte yapılacak şekilde plan oluşturarak, uçağın tekrar hangara alınmasından ve bakım için ek ücret ödenmesinden işletmeleri kurtarmış oluruz. İşlemleri birlikte veya ayrı yapma kararı kullanıcıya aittir, oluşturduğumuz plan bir "öneridir", çünkü gerçek hayatta planlı bakımların günleri ve saatleri sefer iptali, gecikmeden veya bakım sırasında kullanılacak parça eğer şirketin

deposunda yok ise bakımları erteleyebilirler. Bu durumda, sistem motor bakım/söküm tarihine 1.5-2 ay kala tüm bakımları inceleyerek en uygun bakımı bularak kullanıcıya son durumu bildirmektedir. Sistem sadece kendi bakımlarını değil, aynı tipte olan diğer uçakların bakımlarını ve motor bakım sökümlerini incelemektedir. Kullanıcı isterse onaylar, isterse farklı bir bakım planı ile birlikte de motor bakım/sökümünü yapabilir. Eğer X ve Y uçağın motoru bakıma girecekse, öncelikle nedenine bakılır, varsayalım X uçağa takılı motor ömrü bittiği için değişecek, fakat Y uçağı tamamen ağır bakım görecek ve 90 gün boyunca hangarda kalacak, ama motorun daha ömrü var ise, bu durumda Y uçağından motor sökülerek X'e takılır, Y uçağı hem ağır bakıma girer aynı anda X'den sökülerek Y'ye gelen motor bakım görür. Yeni nesil motorlarda, 'Bleed valve, Low pressure turbine blade veya 5th bearing' gibi parçaların ömrü bitti veya arızalandı ise, motoru indirmeden bakım yapılabilmektedir.

Bir kaç örnek senaryo oluşturarak elde edilen sonuçlar aşağıda verilmiştir.

Tablo 3.10 Örnek Senaryo 1

Ay	1	2	3	4
Servis dışı	A(A1)		B(B1&B2)	C(C2)
Servis başlangıcı	A(B2)		B(A1&C2)	C(B1)

Örneğin: 1. ayda A uçağın 1 motorunun, 2. ayda B uçağının komple ağır bakıma gireceğini, 4. ayda ise C uçağının 2ci motorunun bakıma gireceğini düşünürsek, böyle bir senaryoda, B uçağı komple bakıma gireceğinden, 1. ayda B uçağın bir motoru sökülerek A ya takılır, eğer C uçağının rotası aynı bakım istasyonuna denk gelirse aynı anda diğer motor C'ye verilir, B uçağına ise ömrü yaklaşan A ve C uçakların motorları takılır, Eğer C uçağı A ve B uçağının motor değişim bakım istasyonuna denk getirilemezse 3. ayda B uçağı ağır bakıma girdiğinde, motor değişikliği yapılır. Benzer motor değişiklik işlemi en fazla 1 gün almaktadır. Bu senaryo ile bakıma gönderilecek uçağı zamanında göndererek, kalan 2 uçak ile operasyonun devam etmesi sağlanır. Eğer verilen aralıkta ağır bakıma girecek B uçağı olmasaydı, sistem depoda kullanabilir motor olup olmadığını kontrol ederek öneride bulunacaktı. Depoda kullanılabilir bir motor yok ve mevcut operasyon

yapan uçaklar arasında motor değişikliği yapılamıyorsa, kullanıcıya ilgili tip motor arayışına girmesini ve kiralama gerektiği önerilir. Tüm kontrollerin arama alanı 1.5-2 ay tutulmuştur (motor kiralanacaksa, bulunarak sipariş verilme ve tedarikçiden depoya giriş yapma süreci asgari 1 aydır).

Tablo 3.11 Örnek Senaryo 2

Ay	1	2	3	4
Servis dışı	A(A1), E(E1)		B(B1&B2), C(C2)	D(D1)
Servis başlangıcı	A(A1),E(D1)		B(B2&C2), C(B1)	D(X1)

Performans kaybı ve arıza tespitinin motorun bakıma girmesine veya sökülme işleminde önemli yer aldığını daha önceki bölümde vurgulamıştık. 1 ayda A ve E uçaklarında arıza tespit edilirse, sistem öncelikle girilen arızaların analizini yaparak ilgili arızalardan dolayı motorun sökülüp sökülmediğine bakar (geçmiş verilere ve uzman kişinin bilgilerine dayanarak kontroller yapılır), A uçağında çıkan arıza motorun sökülmeden hat bakımında onarılabilecekse operasyonuna devam ettirilir, E uçağında ağır hasar tespiti yapılırsa, sonraki aylarda bakıma girecek en uzak ve ömrü olan motoru bularak (uçuşlar baz alınarak aynı bakım istasyona denk gelecek uçak seçilir) uçaklar arası değişim yapılır, senaryoda D uçağından motor sökülerek E'ye takılır, D uçağında ise 1. senaryoda kullandığımız fonksiyon kullanılır, uygun motor olup olmadığına bakılır ve elde olan bir motor takılır (depoda en az 2 boş motor bulundurulmak zorundadır). 3. ayda C2 motorumuz performans kaybından bakıma girecek ve B uçağımız ağır bakıma gireceğinden, B1 motorumuzun ömrü olduğundan C1 ile yer değiştirilerek C uçağının operasyona devam ettirilmesi sağlanır. Ariza tespiti motor bakım planlama için en kritik noktalardan biridir, ne zaman ve kaç uçakta çıkacağını bilemediğimizden, sistem ortalama arıza tespit değerine bakarak, kullanıcıya belirli dönemlerde depoda ek motor bulundurması uyarısı yapar. Genellikle, yazın uçaklar çok operasyon yaptıklarından dolayı, yaz dönemlerinde bu kontrol şirketlere avantaj sağlamaktadır.

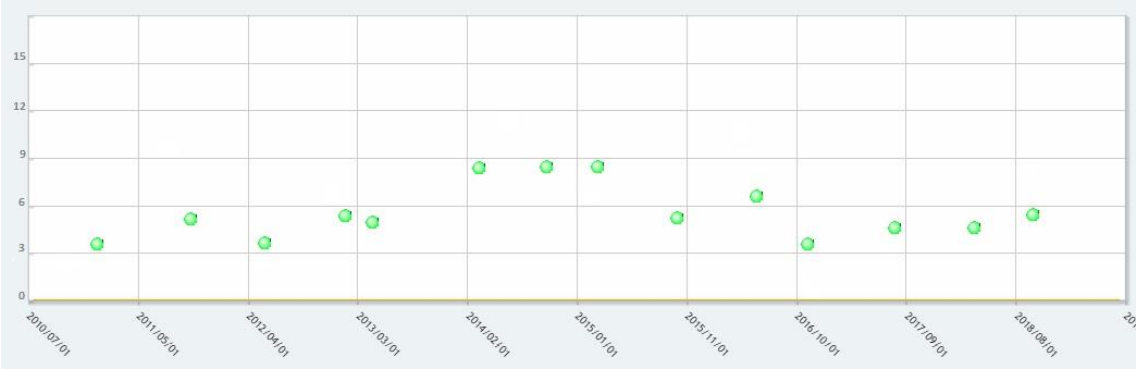
Kiralama iadesi motor sökme işleminde önemli bir kriterdir. Örnek 1 ve Örnek 2 gibi kontroller yapılarak, iade edilecek motor indirilir, yerine uygun motor

bulunarak takılır. Kiralama iade tarihi değişmediğinden, şirketler kira bitimine 1-2 ay kala yeni motor siparişinde bulunduklarından planlaması kolaydır. Sistem girilen tarihi baz alarak, tarih yaklaşınca kullanıcıyı uyarır.

Yabancı bir havayolunun son 10 yılda gerçekleşen motor söküm planı Şekil 3.17 gösterilmiştir. Gerçekleştirilen KDS tarafından aynı şirketin verileri ile oluşturulan söküm planı ise Şekil 3.18'da yer almaktadır. Uygulamanın ortalama olarak sökülmesini gerekli gördüğü motor sayısı şirketin gerçekleştirdiği duruma göre daha optimum seviyededir.



Şekil 3.17 Gerçek Motor Söküm Planı



Şekil 3.18 Hesaplanan Motor Söküm Planı

3.7.4 Uygulama

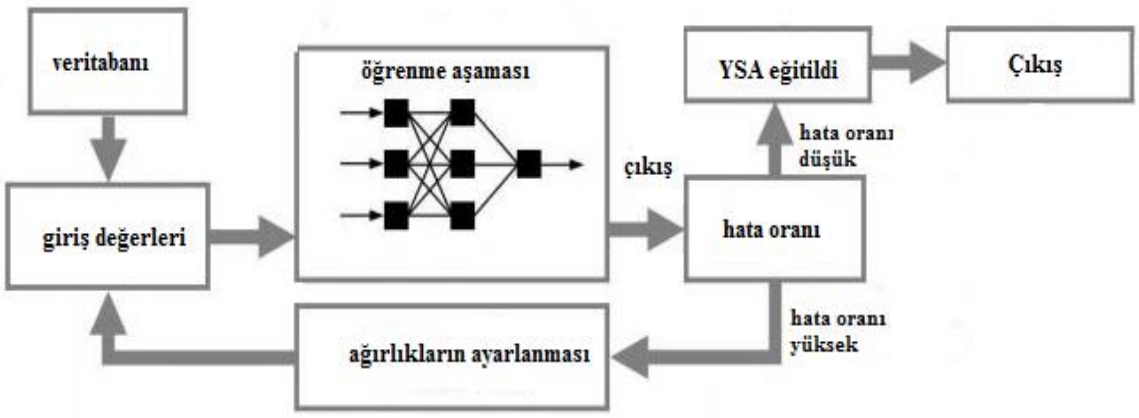
Elde edilen ölçülebilir ve ölçülemeyen parametreleri ağa göndermeden önce tüm parametreler belirlenen sınırlar altında kalıyor mu diye kontrol edilir. Örneğin: çatlak 1-2 mm, yani 5 mm aşmıyor ise, söküm işlemini etkileyemeyeceği için ilgili parametre giriş parametresi olarak hesaba alınmaz. Aynı şekilde N1 hız ve EGS değerleri belirlenen sınır altında ise, giriş parametresi olarak ağa

gönderilmemekte. En önemli sınır verisi, parçanın kalan ömür değeridir, 90 devir altında kalan motorlar direk hesaba alınmaktadır. Parametreleri sınır değerleri Tablo 3.13 verilmiştir.

Tablo 3.12 Parametreleri Sınır Değerleri

Parametre	Sınır Değeri
1st run	≥ 20000
2nd run	≥15000
3rd run	≥ 10000
İade Tarihi	≤ 90
LLP	≤ 90
N1 Hız	≥ 105
Rotor Hızı	≥ 95
Boroskop Sonucu	≥ 3-5 cm

Ağın öğrenme süreci, uzman değerlendirme ile öğrenme örneklerini ağa sunulması ile gerçekleşir. Her bir örnek ağın giriş birimine gönderilir. Ardından veriler ağ yapısı içinde işlenerek ağın çıkış değeri hesaplanır. Elde edilen çıkış değeri ağın istenen çıkış değeri ile karşılaştırılır. Daha sonra, ağın hata değeri hesaplanır. Hesaplanan hata oranı belirlenen hata oranından büyük ise, ağdaki bağlantıların ağırlıkları ağırlık fonksiyonuna (W_Update) bağlı olarak değişir. Hatalar tekrar hesaplanır ve tüm öğrenme süresi boyunca hata kabul edilebilir derecede düşük bir seviyeye ulaşana kadar her nöron için ağırlıklar ayarlanır.



Şekil 3.19 Ağ işleyişi

$$z_i = f(y_i) = f(w_{i0} \cdot x_0 + w_{i1} \cdot x_1 + \ldots + w_{im-1} \cdot x_{m-1}) \quad (3.11)$$

İlgili formül ağırlık-parametre çarpımının tanımıdır. Kısaca Y = W · X, olarak gösterilebilir. Burada W matrisin ağırlığı, X giriş değeri ve Y çıkış değeri.

Geçmiş verileri ağdan geçirerek doğruluk oranı ve hata değerini hesapladıktan sonra bir sonraki veriler bu hata değerin altında kalacak şekilde veriler ağdan tekrar geçirilir. Hata hesabında, işlevi en aza indirmek için, bir sonraki adım w^{n+1} koşulu yerine getirmelidir [85]:

$$f(w^{n+1}) < f(w^n) \quad (3.12)$$

Yani, w^{n+1} noktasındaki f fonksiyonunun değeri, önceki w^n adımındaki fonksiyonun değerinden düşük olmalıdır.

Aşağıda verilen matriste sırayla mbo, n1, egs, korozyon, arıza, hasar ve kalan ömür verileri mevcuttur. Diğer motor parametreleri belirlenen sınır değerleri haricinde kaldığı için giriş parametresi olarak alınmamıştır. Örnek parametre tablosu:

680	98.6	72	3	7	0	90
685	100.4	68	0	5	2	90
650	99.5	72	0	10	0	90
692	100.2	66	1	4	1	90
674	100.8	50	3	7	1	90
660	98.6	49	2	11	0	90
685	101.4	49	2	6	2	90
660	99.2	51	0	7	0	90
685	98.6	48	0	9	1	90
689	100.8	46	1	10	1	90

Şekil 3.20 Örnek Parametre Değerleri Matrisi

Rastgele üretilen ağırlık matrisi (Şekil 3.21) ile Sekil 3.20'de verilen matris çarpılır ve Şekil 3.22'de verilen ağırlık değerleri elde edilir (ağırlık matrisi -0.5 ve 0.5 sayılarla doldurulur). Bu çarpma işlemi belirlenen hata değeri altında kalıncaya kadar devam edilir.

0.5	0.1
−0.2	−0.4
−0.1	0.2
0.3	0.5
0.4	0.5
0.5	−0.5
−0.3	0.3

Şekil 3.21 Ağırlık Matrisi

W_{11} = 680*0.5+98.6*(-0.2)+72*(-0.1)+3*0.3+7*0.4+0*0.5+90*(-0.3) = 289.78

W_{12} = 680*0.1+98.6*(-0.4)+72*0.2+3*0.5+7*0.5+0*(-0.5)+90*0.3 = 74.96

W_{21} = 685*0.5+100.4*(-0.2)+68*(-0.1)+0*0.3+5*0.4+2*0.5+90*(-0.3) = 291.62

W_{22} = 685*0.1+100.4*(-0.4)+68*0.2+0*0.5+5*0.5+2*(-0.5)+90*0.3 = 70.44

289.78	74.96
291.62	70.44
274.90	71.6
294.76	71.32
289.04	68.58
283.38	69.86
294.32	67.74
280.86	67.02
295.08	69.66
297.54	69.78

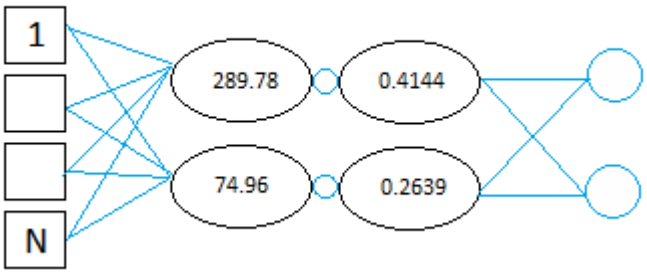
Şekil 3.22 Hesaplanan Ağırlık Matrisi

Aktivasyon kısmın denklemi (2 katman):

output[i] = (1 / (1 + Math.Exp((y * -1))));

o_1 = 1/(1+exp(-289.78)) = 0.414420276

o_2 = 1/(1+exp(-74.96)) = 0.263994743



Şekil 3.23 Hata Değerleri

Katman 3:

Z_3 = ((-0.1)* 0.414420276 + 0.3*0.263994743) = 0.037756395

Z_4= (0.5 * 0.414420276 + (-0.2)*0.263994743)) = 0.154411189

Z_5 = (0.5 * 0.509437977+ 0.5 * 0.538526279)) = 0.523982128

o_3 = 1/(1+exp(-0.037756395)) = 0.509437977

o_4 = 1/(1+exp(-0.154411189)) = 0.538526279

o_5 = 1/(1+exp(-0.523982128)) = 0.628078448

Hata hesabı:

δ_4 =0.538526279*(1-0.538526279)*(0.5*(0.132044901))=0.016407617

δ_5 = (0,9-0.628078448)* 0.628078448*(1-0.628078448)=0.132044901

Ağırlık hata hesabı aşağıdaki gibi hesaplanır:

W_4 = 0.5+(0,9*(0.132044901)) * 0.538526279=0.563998684

W_5 = 0.5+(0,9*(0.132044901)) * 0.628078448=0.5746411

İleri fonksiyonun hesabı aşağıda verilen denklem ile yapılır [85]:

İlk olarak $s = \sum_{i=1}^{n} x_i w_{ij}$, ardından a= $\frac{1}{1+\exp(-s)}$ denklemi kullanılır.

a – Aktivasyon;

δ_j= J dizinde eleman hatası;

K - Katman sayısı;

S - Kombine edilmiş giriş değeri.

Geri yayılım fonksiyonu ise [85]:

İlk olarak $\sigma_j(1-\sigma_j) \sum_k \delta_k w_{kj}$, ardından a=$\sum_k \delta_k w_{kj}$ denklemi ile hesap edilir.

Ağın sonucunu ve hatasını hesapladıktan sonra geri yayılım fonksiyonu hesaplanır. Geri yayılım algoritması çıkış nöronundan başlar. Çıkış nöronun sinapsları olmadığı için aşağıdaki denkle kullanılır:

$$\delta_o = (olması\ gereken\ çıkış - elde\ edilen\ çıkış) * \sum_{i=1}^{n} x_i w_{ij} \qquad (3.13)$$

Diğer katmandaki nöronlar için ise:

$\sigma_j(1-\sigma_j) \sum_k \delta_k w_{kj}$, formül ile hesap yapılır.

Ardından gradiyent değeri aşağıdaki denklem ile hesap edilir:

Grad = σ_j * output.

Ağırlığı değiştirmek için kullanılan denklem (eps-öğrenme hızı) [86]:

$$\Delta w_i = Eps * GRADw + \alpha * \Delta w_{i-1} \quad (3.14)$$

Eps = 0.7;

$\alpha = 0.3$;

$w_5 = 1.5$;

$GRADw_5 = 0.09$;

Δw_5 (i-1) = 0 olduğunda,

Δw_5 = 0.7 * 0.09 + 0 * 0.3 = 0.063;

w_5 = w_5 + Δw_5 = 1.563

Gerçekleştirdiğimiz hesaplamaya göre w_5 ağırlığı 0.063 artacaktır. Diğer katmanlarda aynı hesabı yaparak ağırlıklar güncellenir ve tekrar ileri yayılım fonksiyonu çağırılarak hata oranı istediğimiz orana gelene kadar tekrar edilir. Giriş numunesi [0.1 0.9] ve hedef çıkış değeri 0.9 için. Ağın x_n-2-2-1 mimarisi vardır (giriş elemanların sayısı filtrelenen parametrelere göre ayarlanır).

Yapılan işlemler:

1. Giriş parametrelerini al ve ağ çıkış parametresini belirle (0.65). Öğrenme oranı 0,9 (öğrenme oranını ne kadar yüksek tutarsak öğrenme o kadar hızlı gerçekleşir (değer 0-1 arasıdır)). Giriş parametreleri aktivasyon fonksiyonundan geçir.
2. Çıkış katmanı için hata değerin hesaplanması ve çıkış katmanın ağırlık hata hesabın yapılması (δ_4, δ_5 ve w_4, w_5)
3. Geri yayılım fonksiyonu ile katmanların hata oranını hesapla.
4. Ağırlıkları ayarla ve $w_i = w_i - \alpha.(\partial \text{Error}/\partial w_i)$ ile ağırlıkları optimize et.

   W_Update(0.9) olarak çağırılır.
5. Eğitme fonksiyonu çağırılır.
6. Hesaplanan hata değeri belirlediğimiz 0,0000007 değerinden büyük ise, 1. adıma git.

Öğrenme hızını 0.9, epoch sayısını 10000, hata değeri ise 0,0000007 olarak belirlendi. "network.Egitme(x,y,0.9,0,0000007,10000);" şekilde fonksiyonu

çağırarak ağın eğitilmesini sağlandı. 10000 epoch ile sonuç değeri % 98.171 doğruluk elde edildi.

Son birimde, bulanık mantık yöntemi kullanıldı. jFuzzyLogic kütüphane yazılımı kullanarak motor söküm tarih hesaplama işlemi gerçekleştirildi. Elde edilen değer (remove_date) motorun iade tarihi ile karşılaştırılarak, eğer 'remove_date' iade tarihinden küçük ise elde edilen değer sonuç olarak kabul edilir. Değilse, iade tarihi performans tarihi ile karşılaştırılır. Eğer performans tarihi iade tarihinden küçük ise iki tarihi toplayarak ortalaması alınır. Değilse, söküm tarihi iade tarihi olmaktadır.

Buna göre, dilsel değişkenleri olan bir tablo oluşturulur. Bulanık mantıkta, değerleri sayı değil, terimler olarak adlandırılan doğal dilin sözcükleri olan dilsel değişken kavramı ortaya atılmaktadır.

Tablo 3.13 Dilsel Değişkenler

Tarih	Yakın	Normal	Uzak
Söküm Tarihi	Doğru	Orta	Hatalı
Iade Tatihi	Hatalı	Doğru	Orta
Performans Tarihi	Orta	Hatalı	Doğru

Hücrelerde (hatalı, doğru, orta) terimleri verilmıştır. Her terim için üyelik fonksiyonları tanımlanır. Anlam açısından, bunlar 0'dan 1 e kadar "doğruluk fonksiyonları" olacaktır. Her i satırı için, m(i) hücrelerinde maksimum değerleri hesaplanır. Sonra tüm bu m (i) 'den minimum bulunur. Hesaplanan minimum değer cevap olacaktır.

Mantıksal sonucun ilk adımında, kuralın üyelik derecesinin belirlenmesi gerekir. Bunun için bulanık mantıkta iki operatör vardır: MIN (...) ve MAX (...). Birincisi, üyelik derecesinin minimum değerini, ikincisi ise maksimum değerini hesaplar. Hangi fonksiyonun ne zaman uygulanacağı kuralın ilişkili olduğu pakete bağlıdır. AND kullanılıyorsa, MIN operatörü (...) kullanılır. Değerler OR ile birleştirilirse, MAX (...) operatörünü kullanmanız gerekir.

Hesaplardan elde edilenler:

min (maks (0.5, 0.33, 0), maks (0, 0.5, 1), maks (1, 1, 0)) = min (0.5, 1, 1) = 0.5

Hesaplama Kuralları:

Aşağıdaki noktalara dikkat ederek CPU zaman kazandırıldı.

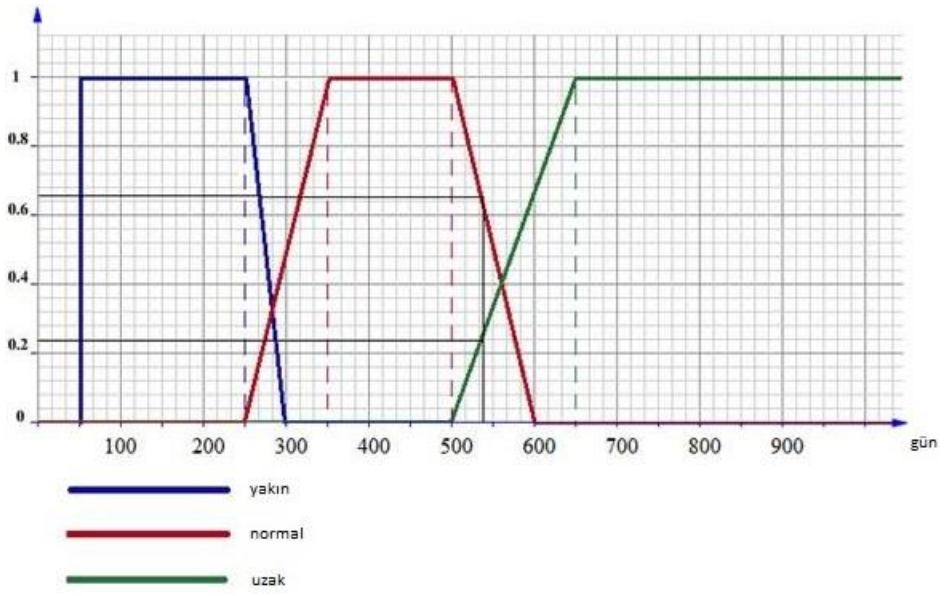
1) Eğer max fonksiyonu için terim hesaplanırsa ve 1'e eşitse, o zaman gerisini hesaplamanın anlamı yoktur, maksimum 1 olur.
2) Min fonksiyonu için terim hesaplanır ve 0'a eşitse, o zaman gerisini hesaplamak mantıklı olmaz, min 0 verecektir.

Dilsel terimlerin her biri için üyelik fonksiyonları oluşturuldu. Yamuk ve s şeklindeki üyelik fonksiyonları kullanarak optimal çözüm hesaplandı. Parametreler aşağıdaki gibi verilir:

Yakın = [50, 50, 250, 300], Normal = [250, 350, 500, 600], Uzak = [500, 650].

Örnek olarak 535 değeri ele alalım, 535 hem normal hem uzak fonksiyonlarında kesişmektedir. En büyük dereceli elemanlardan en küçük olanı (Smallest of Maximum (SoM)) durulaştırma yönetimini kullanarak sonuç bulunur. Bulduğumuz en küçük değer durulaştırma sonucu olarak alınır.

$\mu_{yakın}(535) = 0;\ \mu_{normal}(535) = 0{,}65;\ \mu_{uzak}(535) = 0{,}23;$



Şekil 3.24 Bulanık Mantık Örneği

Tablo 3.14 YSA Başarı Ölçümü

Motor	KDS	Uzman	Hata
A1	26-05-2022	25-05-2022	-1
A2	06-05-2019	06-05-2019	0
A3	07-07-2019	07-07-2019	0
A4	06-12-2023	08-12-2023	2
A5	22-12-2024	23-12-2024	1
A6	06-05-2019	05-05-2019	-1
A7	01-02-2022	01-02-2022	0
A8	04-10-2020	04-10-2020	0
A9	12-06-2020	11-06-2020	-1
A10	30-11-2019	30-11-2019	0
A11	11-11-2019	11-11-2019	0
A12	07-02-2028	05-02-2028	-2
A13	07-09-2025	05-09-2025	-2
A14	12-01-2035	15-01-2035	3
A15	15-03-2029	15-03-2029	0
A16	30-06-2033	29-06-2033	-1
A17	27-01-2035	30-01-2035	3
A18	22-11-2027	23-11-2027	1
A19	22-11-2027	23-11-2027	1
A20	01-06-2032	03-06-2032	2

Ortalama hata oranı 0.2'dir. Kabul edilebilir bir değer olduğundan, yapılan bu çalışmada YSA dinamik KDS'de başarı ile kullanılabileceği görülmüştür.

4
Sonuç ve Öneriler

Yapılan araştırmalar ve çalışmalar göz önüne alındığında, uçakların güvenliği düşünülerek yapılan motor bakım ve söküm işlemlerinde özellikle uçak mühendislerinin bir karar verme destek sisteminden faydalanması oldukça pratik olabileceğinden yola çıkılmıştır.

Problem çözme sürecinde ortaya çıkan zorluk ve karmaşıklık, KDS'in önemini ortaya çıkartır. İlk olarak [9-27]'de sunulan sonuçlara dayanarak, motor ve motor parçalarının bakım planlamasının sonuçlarını tahmin etmek için yapay zekânın unsurları kullanılabilir. Belirsizlik durumunda ise, bulanık mantık algoritması kullanılır. Bu durum, motor takım/söküm süreci hakkında tam bir bilgi yokluğunda, bilinçli bir karar verilmesini mümkün kılar.

Çalışmanın yeniliği, havacılık alanında motor takım/söküm işlemini en optimize bir şekilde sunan karar destek sistemi olmasından kaynaklanır. Araştırmada, motor bakım/söküm işlemini belirleme sürecini otomatikleştirmek için bir karar destek sistemi geliştirilmiştir.

1. Mevcut sistemlerden farklı olarak, Bulanık Mantık ve YSA kullanan KDS modeli önerilmiştir.
2. Uçak motorların takım/söküm tarihlerini en optimize şekilde hesaplanmasına olanak sağlayan sinir ağlarına ve bulanık mantığa dayanan KDS modelini oluşturan yöntem geliştirilmiştir.
3. YSA ve bulanık mantığa dayalı bir karar destek sistem yapısının tasarlanması gerçekleştirilmiştir.

Sonuçların pratik önemi, sinir ağlarına ve bulanık mantığa dayalı bir KDS ile oluşturulan gelişmiş simülasyon modeline dayanmaktadır. Bu çalışmayla, gelecekte yapılması planlanan otomatik bakım programının takip ve oluşturma çalışmasının zeminini hazırlaması öngörülmektedir.

Denetim fonksiyonları oluşturarak, motorların hangi parçalarının denetlenmesi gerektiğini, ne zaman denetlenmesi gerektiğini ve ne kadar sıklıkla denetiminin yapılması gerektiği incelenmiştir. Çalışmada, ölçülebilen ve ölçülemeyen kriterleri değerlendirilerek BÇS ile belirlenen kurallara göre kombinasyon yapıldı. Karmaşık verinin analizinde çıkan farklılıklar bulanık mantık kullanılarak azaltılmıştır. YSA ile kullanılabilir sonuçlar elde edilmiştir. YSA, kriter ve alternatifleri arasında ilişkinin ve kuralların tanımlanmasında kullanılmıştır.

Uygulama sonucunda KDS'nin tutarlı sonuçlar verdiği tespit edilmiştir. Mevcut veri ile Optimizer'da %92, SV-Plan'da %95 doğruluk seviyesinde sonuçlar elde edilirken, geliştirdiğimiz KDS'de %98 doğruluk seviyesinde sonuçlar elde edilebilmiştir. KDS'deki hesaplama, motor bakım/söküm sürecini iyileştirerek işletmelerin karını arttıracağı için büyük önem arz etmektedir.

Çalışmada kullandığımız YSA ve BM yöntemlerin birlikte kullanıldığında sistemin kullanıcıların sorgularına daha hızlı cevap vermesi ve sonuçların doğruluk oranı yüksek olduğu gözlemlenmiştir. Büyük veri setleri sistemin yavaş cevap vermesine sebep olmaktadır. Hızlandırma ve optimize etme konusu üzerinde çalışmalar önerilmektedir.

A
Ekler

I. Giriş birimine gönderilen parametreler:

```
import java.util.*;
public class Vector
{
	public double[] v; // vektör değeri
	public int n; // vektör uzunluğu
	public Vector(int n)
	{
		this.n = n; // uzunluk değerini kopyalıyoruz
		v = new double[n]; // dizi oluşturduk
	}
	public Vector(Double... values)
	{
		n = values.length;
		v = new double[n];
		for (int i = 0; i < n; i++)
		{
			v[i] = values[i];
		}
	}
	public final double get(int i)
	{
		return v[i];
	} // değeri çagırma
	public final void set(int i, double value)
	{
		v[i] = value;
	} // değeri değiştirme
}
```

II. Matris (ağırlık değerlerin tutulduğu kısım):

```
import java.util.*;
```

```
public class Matrix
{
        private double[][] v; // matris değerleri
        public int n, m; // satır ve sütun sayısı
// verilen büyüklükte matris oluşturma ve rasgele sayılarla doldurma (-0.5, 0.5)
        public Matrix(int n, int m, Random rastgele)
        {
                this.n = n;
                this.m = m;
                v = new double[n][];
                for (int i = 0; i < n; i++)
                {
                        v[i] = new double[m];
                        for (int j = 0; j < m; j++)
                        {
                                v[i][j] = rastgele.nextDouble() - 0.5; // rastgele sayılar
                        }
                }
        }
        public final double get(int i, int j)
        {
                return v[i][j];
        } // değeri çağırma
        public final void set(int i, int j, double value)
        {
                v[i][j] = value;
        } // değeri değiştirme
}
```

III. Ağ tanımı.

```
import java.util.*;
import java.time.*;
class Network {
  constructor(sizes) {
    let rast = new Random(DateTime.Now.Millisecond);
    this.Katman = (sizes.Length - 1);
    this.W = new Array(this.Katman);
```

```
    for (let k = 1; (k < sizes.Length); k++) {
      this.W[(k - 1)] = new Matrix(sizes[k], sizes[(k - 1)], rast);
    }
  }
  Ileri(input) {
    let output;
    for (let k = 0; (k < this.Katman); k++) {
      output = new Vector(this.W[k].n);
      for (let i = 0; (i < this.W[k].n); i++) {
        let y = 0;
        for (let j = 0; (j < this.W[k].m); j++) {
          y = (y + (this.W[k][i, j] * input[j]));
        }
        output[i] = (1 / (1 + Math.Exp((y * -1))));
        // uçgen fonksiyonuna gore
       // output[i] = ucgen(y, Math.min(y), Math.max(y));
      }
    }
    return output;
  }
}
```

IV. Öğrenme birimi.

```
import java.util.*;
import java.time.*;
import java.io.IOException;
class Network {
}
{
  x: Vector;
  z: Vector;
  df: Vector;
}
W: Matrix[];
```

```
L: LayerT[];
d: Vector[];
katman: number;
constructor(sizes, number[]);
{
  let rast = new Random(DateTime.Now.Millisecond);
  this.katman = (sizes.Length - 1);
  this.W = new Array(this.katman);
  this.L = new Array(this.katman);
  this.d = new Array(this.katman);
  for (let k = 1; (k < sizes.Length); k++) {
    this.W[(k - 1)] = new Matrix(sizes[k], sizes[(k - 1)], rast);
    this.L[(k - 1)].x = new Vector(sizes[(k - 1)]);
    this.L[(k - 1)].z = new Vector(sizes[k]);
    this.L[(k - 1)].df = new Vector(sizes[k]);
    this.d[(k - 1)] = new Vector(sizes[k]);
  }
}
Ileri(input, Vector);
Vector;
{
  for (let k = 0; (k < this.katman); k++) {
    if ((k == 0)) {
      for (let i = 0; (i < input.n); i++) {
        this.L[k].x[i] = input[i];
      }
    }
    else {
      for (let i = 0; (i < this.L[(k - 1)].z.n); i++) {
        this.L[k].x[i] = this.L[(k - 1)].z[i];
      }
    }
    for (let i = 0; (i < this.W[k].n); i++) {
```

```
        let y = 0;
        for (let j = 0; (j < this.W[k].m); j++) {
          y = (y
            + (this.W[k][i, j] * this.L[k].x[j]));
        }
        this.L[k].z[i] = (1 / (1 + Math.Exp((y * -1))));
        this.L[k].df[i] = (this.L[k].z[i] * (1 - this.L[k].z[i]));
      }
    }
    return this.L[(this.katman - 1)].z;
}
```

V. Geri yayılım (hata hesabı yapılan kısım).

```
// geri yayılım
Geri(output, Vector, error, number);
{
    let last = (katman - 1);
    error = 0;
    for (let i = 0; (i < output.n); i++) {
      let e = (L[last].z[i] - output[i]);
      d[last][i] = (e * L[last].df[i]);
      error = (error + (e * (e / 2)));
    }
    for (let k = last; (k > 0); k--) {
      for (let i = 0; (i < W[k].m); i++) {
        d[(k - 1)][i] = 0;
        for (let j = 0; (j < W[k].n); j++) {
          d[(k - 1)][i] = (d[(k - 1)][i]
            + (W[k][j, i] * d[k][j]));
        }
        d[(k - 1)][i] = (d[(k - 1)][i] * L[(k - 1)].df[i]);
      }
    }
}
```

VI. Ağırlık değişimi.

```
// ağırlık güncelleme, alfa - öğrenme hızı
W_Update(alpha);
{
  int eps = 0.00000001; // düzeltme katsayısı
  int b1 = 0.9;
  int b2 = 0.999;
  int alpha = 0.001
  int L = 10; // tekrarlama
  this.m = new Array(this.katman);
  this.v = new Array(this.katman);
  for (let k = 0; (k < katman); k++) {
    for (let i = 0; (i < W[k].n); i++) {
      for (let j = 0; (j < W[k].m); j++) {
        W[k][i, j] = (W[k][i, j] - (alpha * (d[k][i] * L[k].x[j]))); // 1
m[k][i, j] = ((b1 * m[k][i, j]) + ((1 – b1) * W[k][i, j]));
v[k][i, j] = ((b2 * v[k][i, j]) + ((1 – b2) * Square(W[k][i, j])));
let m_cap = (m[k][i, j] / (1 - Math.Pow(Beta1, L)));
let v_cap = (v[k][i, j]] / (1 - Math.Pow(Beta2, L)));
// agırlıkları guncelle
W[k][i, j] = (W[k][i, j] - (alpha * (m_cap / (Sqrt(v_cap) + eps)))); // 2
      }
    }
  }
}
```

VII. Öğrenme birimi.

```
public final void ucgen(int x, int a, int b)
   {
            int c;
            x += (-a);
            if ((a < 0) && (b < 0))
            {
                  b = -(b - a);
```

```
            }
            else
            {
                if ((a < 0) && (b > 0))
                {
                    b += -a;
                }
                else
                {
                    if ((a > 0) && (b > 0))
                    {
                        b -= a;
                    }
                }
            }
            c = (b / 2.0);
            a = 0.0;
            if (x < c)
            {
                return (x / c);
            }
            else if (x > c)
            {
                return ((b - x) / c);
            }
            return 1.0;
    }

public final void Egitme(Vector[] X, Vector[] Y, double alpha, double eps, int epochs)
{
        int epoch = 1;
        double error;
        do
        {
            error = 0;
            for (int i = 0; i < X.length; i++)
            {
                Ileri(X[i]);
```

```
                    Geri(Y[i], error);
                    W_Update(alpha);
                }
                epoch++;
            } while (epoch <= epochs && error > eps);
}
```

VIII. Bulanık birimi

```
public final void fuzzycontrol() {
int remove_date = 0; int performans_date = 0; double output = 0.0;
if (Eng.e_date[0] == Tarih.return_date) {
   return_date = getMaxValue();
   performans_date = getMaxValue();
   remove_date = getMaxValue();
   if (return_date < remove_date) {
      remove_date = return_date;
   }
}
if (Eng.e_date [0] == Tarih.remove) {
  remove_date = getMaxValue()/3;
  performans_date = getMaxValue();
  }
}
else if (Eng.e_date [0] == Tarih.performans) {
   performans_date = getMaxValue();
   remove_date = getMaxValue()/2;
}
try {
```

```
}

fis.setVariable("remove", remove_date);

fis.setVariable("performans", performans_date);

fis.evaluate();

output = fis.getVariable("output").defuzzify();

} catch (Exception e){

   Messagebox(e.getMessage));

}
```

Kaynakça

[1] A.Sezen "Karar Verme - Problem Çözme", 4 Ekim 2012, Erişim Tarihi: 10 Nisan 2019, http://www.bilka.org.tr/c/ detay.asp?id=6811

[2] H. Jantan, A.R. Hamdan and Z.A.Othman, "Intelligent Techniques for Decision Support System in Human Resource Management", Malaysia, 2010

[3] P. G. W. Keen, "Decision support systems: an organizational perspective". Reading, Mass., Addison Wesley Pub. Co. ISBN 0-201-03667-3, 1978

[4] R.Sprague, "A Framework for the Development of Decision Support Systems." MIS Quarterly. Vol. 4, No. 4, pp.1-25. 1980

[5] Power, Daniel, J. "Decision Support Systems; Concepts and Resources for Managers". Greenwood Publishing Group Inc.,2002

[6] M. LAMBA, "Karar Vermeyi Etkileyen Yapısal Faktörler Bakımından Yönetim ve Organizasyon Teorilerinin İncelenmesi", KMÜ Sosyal ve Ekonomik Araştırmalar Dergisi 16 (27): 8-18, 2014 ISSN: 2147 - 7833

[7] F. Gallegos, "Decision Support Systems: An Overview. Information Strategy". The Executive's Journal, 15 (2): 42-46., 1999

[8] Holsapple, Clyde W. and Whinston Andrew B., "Decision Support Systems A Knowledge Based Approach". New York: West Publishing., 1996

[9] Turban, E. "Decision Support and Expert Systems: Management Support Systems" Second Edition. Charlotte: McMillan Publishing,1990

[10] Khaled F. Abdelghany, "An Integrated Decision Support Tool for Airlines Schedule Recovery During Irregular Operations", 2008

[11] Tekiner, H., Birbil, Ş.İ. and Bülbül, K. 2009. "Robust crew pairing for managing extra fligts, Computer Operations Research". (36), 2031-2048.

[12] Armacost, A, Barnhart, C. and Ware, K. , "Composite variable formulations for Express shipment service network design, Transportation Science". (36), 1-20, 2002

[13] Daskin, M.S. and Panayotopoulos, N.C. "A lagrangian Relaxation Approach to Assigning Aircraft to routes in Hub and Spoke Networks, Transportation Science". (2), 91-99, 1989

[14] Feo, T.A. and Bard J. F. "Flight Scheduling and Maintenance Base Planning, Management Science". (35), 1415-1432, 1989

[15] Kabbani, N.M. and Patty B.W. "Aircraft Routing at American Airlines", Proceeding of the 32nd Annual Symposium of AGIFORDS, Budapest, Hungary, 1992.

[16] Clarke, L.W., Hane, C.A., Johnson, E.L. and Nemhauser, G.L. "Maintenance and Crew Considerations In Fleet Assignment, Transportation Science". (30), 249–261, 1996.

[17] S. Bratu, C. Barnhart,"Flight operations recovery: New approaches considering passenger recovery", 2006

[18] Gopalan, R. and Talluri, K. "The Aircraft Maintenance Routing Problem, Operations Research". (46), 260–271, 1998.

[19] Cordeau, J.F., Stojkovi´c, G., Soumis F. and Desrosiers, J. Benders,"Decomposition for Simultaneous Aircraft Routing and Crew Scheduling, Transportation Science". (35), 375–388, 2001.

[20] TAV Journal, Erişim Tarihi: 15 Mart 2019, http://www.tavyatirimciiliskileri.com/tr-TR/Lists/Annual%20Reports/Attachments/14/ TAV_FRAT_SON.pdf

[21] Sriram, C. and Haghani, A. "An Optimization Model For Aircraft Maintenance Scheduling And Re-Assignment, Transportation Science". (37), 29-48, 2003.

[22] Sarac, A., Batta, R. and Rump, C.M. "A Branch-and-Price Aprroach For Operational Aircraft Maintenance Routing", European Journal of Operational Research. (175), 1850-1869, 2006.

[23] Orhan, İ. "Uçak Bakım Planlamasının Optimizasyonuna Yönelik Bir Karar Destek Sistemi", Doktora Tezi, Anadolu Üniversitesi, 2007.

[24] Ryan, D. M. "The solution of massive generalized set partitioning problems in aircrew rostering", The Jornal of the Operational Research Society. (43), 459-467, 1992.

[25] Özdemir, U. "Methodology for Crew-Pairing Problem In Airline Crew A Scheduling", Yüksek Lisans Tezi, Boğaziçi Üniversitesi, Türkiye, 2009

[26] Arikan, U., Gurel, S., Akturk, M.S. "Flight network-based approach for integrated airline recovery with cruise speed control". Transportation science. pp. 1-29, 2017

[27] Tammineni, S.V., Scanlan, J.P. and Reed, P.A.S. "Expert decision support system for costing of aircraft gas turbine engine". In Proceedings of GT-2007 ASME Turbo Expo 2007: Power for Land, Sea and Air. American Society Of Mechanical Engineers (ASME), 2007

[28] J.T.Luxhoj, T.P.Williams,"Integrated decision support for aviation safety inspectors",Finite Elements in Analysis and Design 23,381-403,1996

[29] S.Maynard,D.Arnott,F.Burstein," A Method For Multiple Criteria Evaluation Of DSS In A MultipleConstituency Environment",1999

[30] N. Duerksen,"Fuzzy Logic Decoupled Longitudinal Control for General Aviation Airplanes", 1997

[31] S. N. Balakrishnan "Fuzzy Logic in Restructurable Flight Control Systems", Missouri University of Science and Technology

[32] Kelly, W., Painter, J., "Soft Computing in the General Aviation Cockpit", Proceedings of the First On-line Workshop of Soft Computing, Japan, 151-156, 1996.

[33] A.Nighat Khizer, D. Yaping, S. Amjad Ali and Xu Xiangyang,"Stable Hovering Flight for a Small Unmanned Helicopter Using Fuzzy Control",Article ID 208253 China, 2014

[34] Boskoski, P., Mileva, B., Deskoski, S. "Auto Landing Using Fuzzy Logic", 6th International PhD Workshop on Systems and Control, Izola, Slovenia, 2005.

[35] Meyer, D., Sághi, B., Tarnai, G., "Safety Management Of Traffic Growth in Air Transportation", Transportation Engineering, 36/1-2, 69-72, 2008.

[36] Y. Özdemir, "Aircraft selection using fuzzy ANP and the generalized choquet integral method", Yildiz Teknik Üniversitesi,2016

[37] Vijayan, V. P., John, D., Thomas, M., Maliackal, N. V., Vargheese, S. S., "Multi Agent Path Planning Approach To Dynamic Free Flight Environment", International Journal of Recent Trends in Engineering, Vol 1, No. 1., 2009.

[38] Russian aviation journal, Erişim Tarihi: 24 Nisan 2019, https://tjournal.ru/flood/55010-aircraft-software

[39] Oracle Applications, Maintenance Cloud Software, Erişim Tarihi: 03 Mayis 2019, https://www.oracle.com/applications/supply-chain-management/solutions/maintenance-management/complex-maintenance-repair-overhaul.html

[40] Aircraft IT, Maintenance and MRO Operations, Erişim Tarihi: 02 Mayis 2019 https://www.aircraftit.com/operations/Vendors/Aviation42-2/Modules/Software-for-Aviation.aspx

[41] J.B.Loiseau,"Rosenblatt's perceptron, the first modern neural network",2017

[42] C.Borgelt, "Hopfield Networks and Boltzmann Machines", Computational Intelligence, 2017

[43] Efraim Turban; Jay E. Aronson; Ting-Peng Liang "Decision Support Systems and Intelligent Systems". p. 574, 2008

[44] S.Marre, K.Gibert, "Evolution of Decision Support Systems", University of Catalunya, 2012

[45] Tariq, Rafi,Intelligent "Decision Support Systems- A Framework", India, 2011

[46] Merkert, Mueller, Hubl, "A Survey of the Application of Machine Learning in Decision Support Systems", University of Hoffenhaim 2015

[47] Decision Support Systems (DSS), 7 Şubat 2011, Erişim Tarihi: 15 Mayis 2019, https://www.mbaknol.com/management-information-systems/decision-support-systems-dss/

[48] Jang, J.-S. R., "ANFIS: Adaptive-Network-based Fuzzy Inference Systems, " IEEE Transactions on Systems, Man, and Cybernetics, Vol. 23, No. 3, pp. 665—685, May 1993.

[49] Haettenschwiler, P. (1999). New user-friendly concept of Decision support. Good decisions in business, politics and society. Zurich, vdf Hochschulverlag AG: 189-208.

[50] Hackathorn, R. D., and P. G. W. Keen.. "Organizational Strategies for Personal Computing in Decision Support Systems." MIS Quarterly, Vol. 5,No. 3, September 1981

[51] Gökçen, Hadi, "Yönetim Bilgi Sistemleri", Palme Yayıncılık, Ankara, 2007

[52] J. Fox, "Decision Support Systems" ,2001

[53] Marek J. Druzdzel and Roger R. Flynn, "Decisio, n Support Systems" University of Pittsburgh Pittsburgh,2002

[54] M. Santana, "Managerial learning: A neglected dimension in decision support systems", Proc. of the 28th Ann.Hawaii Int. Conf. on System Sciences, Vol. IV, pp. 82-91, 1995.

[55] Uziel Sandler, Lev Tsitolovsky "Neural Cell Behavior and Fuzzy Logic". Springer, 2008. -478 s.

[56] Z. Şen, "Bulanık Mantık ve Modelleme İlkeleri", Bilge Sanat Yapım Yayınevi, 2001

[57] C. Szegedy,A. Toshev,D. Erhan, «Neural Information Processing Systems», Deep Neural Networks for Object Detection, 2013.

[58] S. Haykin , "Neural Networks: A Comprehensive Foundation" Second Edition

[59] Öztemel, E. 2006. "Yapay Sinir Ağları", ISBN no: 975-6797-39-8, Papatya Yayıncılık, İstanbul.

[60] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," ,1982, pp. 2554–2558.

[61] IBM Tivoli OMEGAMON XE and Tivoli Management Services on z/OS, Erişim Tarihi: 11 Mayis 2019,https://www.ibm.com/support/knowledgecenter/SSAUBV/com.ibm.omegamon_share.doc_6.3.0.1/tmsz63fp1_zcommonconfig.pdf

[62] The Vaisala CheckTime system helps to reduce chemical fluids use in aircraft de-icing, and makes airline traveling smoother 23.02.2017, Erişim Tarihi: 11 Mayis 2019,https://www.editori.fi/artikkeli/the-vaisala-checktime-system-helps-to-reduce-chemical-fluids-use-in-aircraft-de-icing-and-makes-airline-traveling-smoother/

[63] M.Andolin, Towards an intelligent decision support system in adult education April 10, 2018, Erişim Tarihi: 12 Mayis 2019, https://blogs.uta.fi/edumap/2018/04/10/challenges-with-developing-idss-for-ae

[64] Dubai created a Decision Support System based on the integration of the latest weather technology, forecasting extreme weather events and protecting the city's coastline at the same time.2016, Erişim Tarihi: 11 Mayis 2019, https://goexplorer.org/dubai-decision-support-system-sends-extreme-weather-warnings/

[65] Geetha, A. ve Nasira, G.M. "Data Mining for Meteorological Applications: Decision Trees for Modeling Rainfall Prediction" IEEE International Conference on Computational Intelligence and Computing Research, 2014

[66] Zadeh, L. A., "Fuzzy sets," Information and control, 8(3), 338-353, 1965

[67] K. L. Choy, H. K. H. Chow, W. B. Lee, and F. T. S. Chan, "Development of performance measurement system in managing supplier relationship for maintenance logistics providers", 2007

[68] A.S.Walia," Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent", 10 June 2017, Erişim Tarih: 11 Mayis 2019, https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f

[69] Duchi, J., Hazan, E., & Singer, Y. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". Journal of Machine Learning Research, 12, 2121–2159,2011

[70] Kingma, D.P. J.L. Ba. "Adam: A Method for Stochastic Optimization" , 2014.

[71] C. Sriram, A. Haghani, "An optimization model for aircraft maintenance scheduling and re-assignment" , Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20742, USA

[72] N. B. Hölzel,C. Schröder,T. Schilling and V. Gollni "A Maintenance Packaging and Scheduling Optimization Method for Future Aircraft",January 2012

[73] Jacobs, F.R. Playing the Beer Distribution Game over the Internet. Production and Operations Management, 2000, 9(1), 31-39.

[74] IATA, Erişim Tarihi: 11 Mayis 2019, https://www.turkcebilgi.com/iata

[75] R. Zaher "What is the Difference Between FAA, CAA, ICAO, EASA and JAA?", Dec 10, 2015, Erişim Tarihi: 11 Mayis 2019, https://www.paramountbusinessjets.com/blog/what-is-the-difference-between-faa-caa-icao-easa-and-jaa/

[76] Mamdani,E.H., 'Application of Fuzzy Algorithms for Control of Simple Dynamic Plant", Proc.IEEE,1974

[77] Ross, T.J. "Fuzzy Logic With Engineering Applications(Third Edition)". John Wiley&Sons Ltd.,2010

[78] Wang, C. "A Study Of Membership Functions On Mamdani-Type Fuzzy Inference System For Industrial Decision-Making". Lehigh University.(Paper:1665),2015

[79] N. Guarino, Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

[80] Angehrn, Albert A. ve Jelassi, T. "DSS Research and Practice in Perspective, Decision Support Systems", 12 (4-5): 267-175,1994

[81] Noor Maizura Mohamad Noor and Rosmayati Mohemad (2010). New Architecture for Intelligent Multi-Agents Paradigm in Decision Support System, Decision Support Systems, Chiang S. Jao (Ed.), InTech, DOI: 10.5772/39463. Erişim Tarihi:11 Mayis 2019, http://www.intechopen.com/books/decision-support-systems/new-architecturefor-intelligent-multi-agents-paradigm-in-decision-support-system

[82] Bonifati, A., "Designing data marts for data warehouses",2001

[83] Genetik Algoritma - Genetic algorithm, Vikipedi - ansiklopedi, Erişim Tarihi: 11 Mayis 2019 https://tr.qwerty.wiki/wiki/Genetic_algorithm

[84] A. Tortum, N. Yayla, M. Gökdağ. "Yapay sinir ağları ve birleştirilmiş sinirsel bulanık sistemler ile şehirler arası yük taşıması tür seçiminin modellenmesi.",2015

[85] M. Buscema, "Back Propagation Neural Networks", Substance Use & Misuse, 33(2), 233–270, 1998

[86] Z. Hocenski, M. Antunovic, and D. Filko "Accelerated Gradient Learning Algorithm for Neural Network Weights Update", 2010, University J.J. Strossmayer