



A comparative predictive maintenance application based on machine and deep learning

Ayşenur Hatipoğlu^{1,2*}, Yiğit Güneri², Ersen Yılmaz¹

¹Department of Electric-Electronic Engineering, Faculty of Engineering and Architecture, Bursa Uludag University, 16059, Bursa, Türkiye

²Turkish Aerospace Inc., Uludag University R&D Center, 16240, Bursa, Türkiye

Highlights:

- A comparative predictive maintenance application based on machine and deep learning has been developed.
- Predictive Maintenance dataset from UCI Machine Learning Repositories was used.
- The best prediction result was obtained using the Random Forest method.

Keywords:

- Predictive Maintenance
- Machine Learning
- Deep Learning
- Classification

Article Info:

Research Article

Received: 19.12.2022

Accepted: 27.05.2023

DOI:

10.17341/gazimmfd.1221105

Correspondence:

Author: Ayşenur Hatipoğlu
e-mail: aysenur.hatipoglu@tai.com.tr
phone: +90 533 680 5460

Graphical/Tabular Abstract

In this comparative study, the performances of six machine learning methods and two deep learning architectures are analyzed on a predictive maintenance dataset. The names of the methods and the architectures are given in Figure A. Four measures which are accuracy, precision, recall and F1-Scores are used for the performance evaluation.

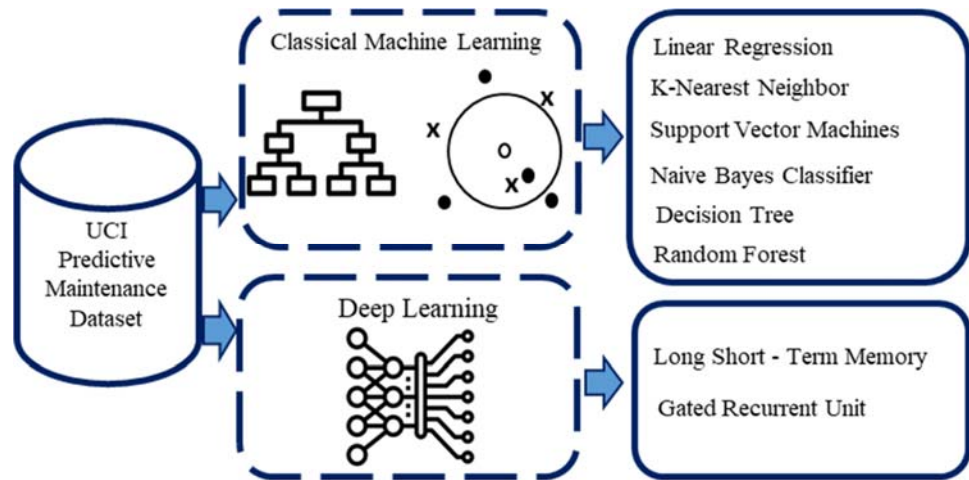


Figure A. Predictive Maintenance System

Purpose:

This study purposed to compare the performance metrics of machine learning-based methods and deep learning architectures for a predictive maintenance application.

Theory and Methods:

Six classical machine learning methods which are Linear Regression (LR), Support Vector Machines (SVM), Decision Tree (DT), Naïve Bayes Classifier, Random Forest (RF), and K-Nearest Neighbor (KNN) and two deep learning architectures namely Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are used for a predictive maintenance application. In the application, the problem is handled as two classification tasks: binary and multi-class. The performances of the methods and the architectures are presented comparatively using accuracy, precision, recall and F1-Scores measures.

Results:

In the multi-class task, the deep learning architectures, LSTM and GRU achieve 97.51% accuracy rate while the classical machine learning method RF reaches the accuracy rate of 98.26%. Binary classification task is done for comparing the performances obtained in this study with those of the existing studies in the literature. For this task RF, LSTM and GRU achieve over 97% accuracy rate.

Conclusion:

Based on the experimental results, it can be seen that classical machine learning methods and deep learning architectures have close performance results on the data set. Therefore, it is concluded that the use of classical machine learning methods for solving the problem is more appropriate than the use of deep learning architectures in terms of both time and cost.



Makine ve derin öğrenme temelli karşılaştırmalı bir öngörücü bakım uygulaması

Ayşenur Hatipoğlu^{1,2*}, Yiğit Güneri², Ersen Yılmaz¹

¹Bursa Uludağ Üniversitesi, Elektrik-Elektronik Mühendisliği Bölümü, Mühendislik Fakültesi, 16059, Bursa, Türkiye

²Türk Havacılık ve Uzay Sanayii A.Ş., Uludağ Üniversitesi Ar-Ge Merkezi, 16240, Bursa, Türkiye

ÖNEÇIKANLAR

- Makine ve derin öğrenme temelli karşılaştırmalı bir öngörücü bakım uygulaması geliştirilmiştir
- UCI Makine Öğrenmesi Ambarlarından alınan Öngörücü Bakım veri seti kullanılmıştır
- En iyi tahmin sonucu Rastgele Orman yöntemi kullanılarak elde edilmiştir

Makale Bilgileri

Araştırma Makalesi

Geliş: 19.12.2022

Kabul: 27.05.2023

DOI:

10.17341/gazimmfd.1221105

Anahtar Kelimeler:

Makine öğrenmesi,
derin öğrenme,
öngörücü bakım,
sınıflandırma

ÖZ

Günümüz teknolojinin veri odaklı endüstriyel sistemlerini potansiyel arızalar oluşmadan önce takip etmek zorunlu hale gelmiştir. Öngörücü bakım bu arızaları oluşmadan önce tahmin eder ve oluşacak arızadan korunmak için zorunlu eylemlerin alınmasını sağlar. Bu çalışmada, makine ve derin öğrenme temelli karşılaştırmalı bir öngörücü bakım uygulaması gerçekleştirilmiştir. Klasik makine öğrenmesi yöntemi olarak Lojistik Regresyon, Naive Bayes Sınıflandırıcı, Karar Ağacı, Destek Vektör Makinesi, Rastgele Orman ve K-En Yakın Komşuluk; derin öğrenme mimarisi olarak ise Uzun Kısa Süreli Bellek ve Geçitli Tekrarlayan Birim kullanılmıştır. Yöntemlerin başarımları UCI Makine Öğrenmesi Ambarlarından alınan Öngörücü Bakım veri seti üzerinde hata tipi tespiti için incelenmiş ve sonuçlar karşılaştırmalı olarak ölçütler bazında detaylı bir biçimde sunulmuştur. Deneysel çalışmalarda hata tipi tespiti çoklu ve ikili sınıflandırma problemi biçiminde ayrı ayrı ele alınmıştır. Çoklu sınıflandırma probleminin çözümünde makine öğrenmesi yöntemleri arasında en yüksek doğruluk değeri %98,26 ile Rastgele Orman yöntemi ile elde edilirken her iki derin öğrenme mimarisi ile elde edilen doğruluk değeri %97,51 olarak gerçekleşmiştir. İkili sınıflandırma probleminin çözümünde ise veri dengelemesi yapıldıktan sonra makine öğrenmesi yöntemleri arasında en yüksek doğruluk değeri %95,03 ile yine Rastgele Orman yöntemi ile elde edilirken derin öğrenme mimarileri arasında en yüksek doğruluk değeri Geçitli Tekrarlayan Birim mimarisi ile %93,03 olarak elde edilmiştir.

A comparative predictive maintenance application based on machine and deep learning

HIGHLIGHTS

- A comparative predictive maintenance application based on machine and deep learning has been developed
- Predictive Maintenance dataset from UCI Machine Learning Repositories was used
- The best prediction result was obtained using the Random Forest method

Article Info

Research Article

Received: 19.12.2022

Accepted: 27.05.2023

DOI:

10.17341/gazimmfd.1221105

Keywords:

Machine learning,
deep learning,
predictive maintenance,
classification

ABSTRACT

It has become imperative to monitor the data-driven industrial systems of today's technology before potential failures occur. Predictive maintenance predicts these failures before they occur and takes the necessary action to prevent malfunctions from occurring. In this study a comparative predictive maintenance application which is based on machine and deep learning is realized. Logistic Regression, Naive Bayes Classifier, Decision Tree, Support Vector Machine, Random Forest, and K-Nearest Neighborhood are used as the classical machine learning methods while Long Short-Term Memory and Gated Recurrent Unit are used as the deep learning architectures. The performances of the methods are examined on the Predictive Maintenance dataset from UCI Machine Learning Repository for fault type detection and the results are presented comparatively in terms of metrics in detail. In the experimental studies, fault type detection is handled separately in the form of multiple and binary classification problems. In the solution of the multi-classification problem, the highest accuracy among the machine learning methods is obtained by Random Forest method with 98.26%, while the accuracy value obtained with both deep learning architectures is 97.51%. In the solution of the binary classification problem, after the data balancing, the highest accuracy among the machine learning methods is obtained by Random Forest method with 95.03%, while the highest accuracy among the deep learning architectures is obtained by Gated Recurrent Unit architecture as 93.03%.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *aysenur.hatipoglu@tai.com.tr, yigit.guneri1@tai.com.tr, esen@uludag.edu.tr /
Tel: +90 533 680 5460

1. Giriş (Introduction)

Günümüzde sistemlerin çalışma sürekliliğinin sağlanması, ilgili işlemlerin aksamaması ve gerekli çalışmaların tam zamanında yerine getirilebilmesi için periyodik olarak bakım çalışmaları yürütülmektedir. Teknolojinin gelişmesi ile Önleyici Bakım (Preventive Maintenance, PvM), Düzeltici Bakım (Corrective Maintenance, CM) ve Öngörücü Bakım (Predictive Maintenance, PdM) teknikleri geliştirilmiştir [1]. Bakım maliyetleri işletme maliyetlerinin önemli bir kısmını oluşturmaktadır. Spesifik sektöre bağlı olarak, bakım maliyetleri üretilen malların maliyetinin yüzde 15 ila 60'ını temsil etmektedir. Bakım yönetiminin etkinliğine ilişkin son anketler, tüm bakım maliyetlerinin üçte birinin (her doların 33 senti) gereksiz veya uygunsuz şekilde gerçekleştirilen bakımın sonucu olarak boşa harcılandığını göstermektedir [2].

Tahmine dayalı bir bakım yönetim programını anlamak için önce endüstriyel tesislerde geleneksel yönetim teknikleri olarak kullanılan PvM ve CM anlaşılmalıdır. PvM, periyodik zaman aralıkları ile bakım çalışmalarının yürütülmesine dayanır. Bu teknikte bakım maliyeti yüksektir [3]. CM ise arıza gerçekleştikten sonra yapılan bakım işlemidir ve periyodik bir çalışma planı yoktur [4,5]. PdM, bakım işleminin zamanında yapılabilmesi için geliştirilmiş veri analizi temelli bir tekniktir [6, 7]. Bu teknik gereksiz bakım işlemlerini ortadan kaldırarak, ihtiyaç duyulan zamanda gerçekleştirilmesine olanak sağlar [8]. Böylece emeği, zamanı ve genel maliyeti artıran boş süreçler azaltılmış olur. Makinelerin ömrü iyileştirildiğinden, gereksiz faaliyetlerde fazladan enerji tüketilmez [7]. PdM tekniği petrol ve gaz endüstrisi, üretim, IT, enerji ve havacılık gibi çok geniş bir yelpazede kullanılmaktadır. Bir PdM sistemi algılayıcılardan topladığı verileri kullanarak bakım öngörüsünü gerçekleştirir [9]. Öncelikle Kalan Faydalı Ömür (Remaining Useful Life, RUL) kestirimi yapılır ve sonraki aşamada bakım işleminin ne zaman yapılması gerektiğinin kararı verilir [10, 11]. PdM'in her iki aşamasında Makine Öğrenmesi (Machine Learning, ML) ve bir alt kümesi olan Derin Öğrenme (Deep Learning, DL) yöntemlerinden faydalanılmaktadır [12]. Bu çalışma kapsamında, klasik ML ve DL yöntemlerinin çoklu ve ikili sınıflandırma başarımları Öngörücü Bakım veri seti üzerinde karşılaştırılmıştır.

2. Metotlar (Methods)

Makine öğrenimi genellikle denetimli öğrenme ve denetimsiz öğrenme olmak üzere iki şekilde kategorize edilir (Şekil 1) [13].

Denetimli makine öğrenimi, adından da anlaşılacağı gibi, uzman denetimi ile etiketlenmiş veriler gerektirmektedir [14]. Denetimli Öğrenme yöntemleri iki temel problemin çözümünde kullanılmaktadır [15, 16]:

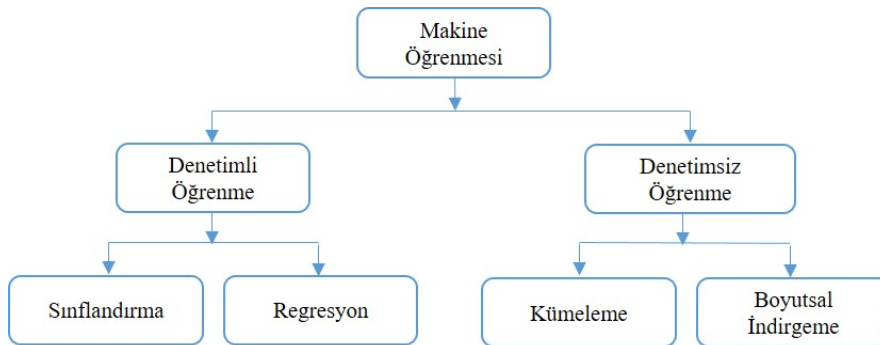
- Sınıflandırma (Classification): Sınıflandırma problemi, önceden tanımlanmış sınıf listesinden bir sınıf etiketinin tahmin edilmesidir. Sınıflandırma, ikili sınıflandırma ve çoklu sınıflandırma olarak iki ayrı grupta ele alınmaktadır [17]. Sınıflandırma yöntemlerine Lojistik Regresyon, Sınıflandırma Ağaçları, Rastgele Orman, Destek Vektör Makineleri ve Yapay Sinir Ağları örnek olarak verilebilir [13].
- Regresyon (Regression): Regresyon problemi, veri setindeki değişkenlerin birbirini nasıl etkilediklerini ve aralarındaki ilişkileri modellemek için kullanılır. Regresyon yöntemlerine Doğrusal Regresyon, Karar Ağaçları ve Bayes Ağları örnek olarak verilebilir.

Sınıflandırma ve regresyon problemleri arasında ayırım yapmanın kolay bir yolu, çıktıda bir tür sıralama veya süreklilik olup olmadığının sorgulanmasıdır. Olası sonuçlar arasında bir sıralama veya süreklilik varsa, problem bir regresyon problemi olarak tanımlanır [16].

Denetimsiz Öğrenme yöntemlerinde etiketli olmayan veri setleri üzerinde, veriler arasındaki ilişki veya gizli kalıplar otomatik olarak belirlenmektedir [17, 14]. Denetimsiz makine öğrenimi yöntemleri etiketli olmayan veriler üzerinde aşağıda verilen temel iki problemin çözümünde kullanılmaktadır [15, 16, 18].

- Kümeleme (Clustering): Kümeleme problemi, veri setindeki etiketsiz verilerin benzer öğelerden oluşan alt gruplarının bulunmasıdır. Kümeleme işleminde, gözlemler arasında homojen dağılıma sahip olan alt grupların tespit edilmesi hedeflenmektedir. Kümeleme yöntemlerine K-Means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [14] ve Hiyerarşik kümeleme örnek olarak verilebilir [19].
- Boyut İndirgeme (Dimensionality Reduction): Boyut indirgeme problemi, verinin yüksek boyutlu özellik uzayından düşük boyutlu bir özellik uzayına indirgenmesidir. Çok sayıda özellik içeren veri kümelerinde makine öğrenmesi yöntemleri ile gerçekleştirilen çözümler uzun süreler alabilmekte ve daha karmaşık modeller gerektirebilmektedir. Bu durumlarda özellik boyutu sayısının indirgenmesi problemin çözümüne katkı sağlamaktadır. Boyut indirgeme yöntemlerine PCA ve IsoMap örnek olarak verilebilir [19, 17].

Bu iki kategori dışında literatürde Yarı Denetimli (semi-supervised) Öğrenme ve Pekiştirmeli Öğrenme (Reinforcement Learning) yöntemleri de yer almaktadır. Yarı denetimli makine öğrenmesi melez bir yöntemdir. Denetimli ve denetimsiz öğrenmenin birleşimi olarak ifade edilebilir. Yarı denetimli makine öğrenmesi yöntemleri, yapısında az sayıda etiketli veri bulundurulur [20, 21]. Bu yöntemler hem etiketlenmiş hem de etiketsiz verileri tahmin işleminde kullanılırlar [22]. Pekiştirmeli öğrenme, sayısal bir ödül sinyali en



Şekil 1. Makine öğrenmesi yöntemleri (Machine learning methods) [13]

üst düzeye çıkarmak için ne yapılacağını, yani durumların eylemlerle nasıl eşleştirileceğini öğrenmektir [23]. Özerk bir ajan, insan operatörün herhangi bir rehberliği olmadan deneme yanılma yoluyla bir görevi gerçekleştirmeyi öğrenir [22]. Yapısında ajan ve çevre olarak adlandırılan fonksiyonlar barındırılır. Diğer temel unsurlar; ilke, ödül sinyali, değer işlevi ve isteğe bağlı olarak bir çevre modelidir [23]. Çevre, ajanın faaliyet gösterdiği fiziksel dünyadır. Bir model, ortamın bir sonraki adımda ne yapacağını tahmin eder. Çevrenin davranışını taklit eder veya daha genel olarak çevrenin nasıl davranacağı hakkında çıkarımlar yapılmasına izin verir. Ödül sinyali çevreden gelen geri bildirimdir. Her zaman adımında çevre, pekiştirmeli öğrenme ajanına ödül adı verilen tek bir sayı gönderir. Ajanın tek amacı, uzun vadede aldığı toplam ödülü maksimize etmektir. Böylece ödül sinyali, ajan için iyi ve kötü olayların ne olduğunu tanımlar. Değer fonksiyonu, bir ajanın belirli bir durumda bir eylemde bulunarak alacağı gelecekteki ödül tahminidir. Durumların iyiliğini/kötülüğünü değerlendirmek için kullanılmaktadır. İlke, en iyi sonuçları elde etmek için ajanın durumunu eylemlerle eşleştirme yöntemidir [23].

Derin öğrenme mimarileri, makine öğrenmesinin bir alt kümesi olup yapay sinir ağları üzerinde temellendirilmiştir. Özellikle denetimsiz veya yarı denetimli öğrenme ile büyük miktarlarda etiketlenmemiş örneklerden faydalanarak başarılı tahminler yapabilmektedirler. Eğitim verilerinin miktarı arttıkça derin öğrenme mimarilerinin kullanımı daha kullanışlı hale gelmektedir [22].

Bu çalışmada, klasik makine öğrenmesi yöntemlerinden Destek Vektör Makinesi, En Yakın Komşuluk, Naive Bayes Sınıflandırıcısı, Lojistik Regresyon, Karar Ağacı, Rastgele Orman ve derin öğrenme mimarilerinden ise Uzun Kısa Süreli Bellek ile Geçitli Tekrarlayan Birim kullanılmıştır. Bu yöntemler ile öngörücü bakım veri seti üzerinde denetimli öğrenme yaklaşımıyla hata tipinin tespiti gerçekleştirilmiştir. Hata tipi tespiti çoklu ve ikili sınıflandırma problemi biçiminde ayrı ayrı incelenmiş ve başarımları geri çağırma, F1-skor, kesinlik ve doğruluk metrikleri kullanılarak detaylı bir biçimde sunulmuştur.

2.1. Klasik Makine Öğrenmesi Temelli Yöntemler (Classical Machine Learning Based Methods)

2.1.1. K-en yakın komşuluk (K-nearest neighbor, KNN)

KNN algoritması sınıflandırma ve regresyon modelleri için kullanılan tekniklerden birisidir. K değeri belirlenerek veri setindeki değişkenlere en yakın noktalar bulunur. K seçimi, elde edilen KNN sınıflandırıcı üzerinde büyük bir etkiye sahiptir [24]. Tahmin sonuçları, özelliklerin nasıl ölçeklendiğine, benzerliğin nasıl ölçüldüğüne ve K değerinin ne kadar büyük olduğuna bağlıdır. Az gürültülü, yüksek düzeyde yapılandırılmış veriler için, daha küçük K değerleri en iyi sonucu verir [25, 26]. KNN algoritmasında verinin k komşularına olan uzaklığı hesaplanırken Manhattan, Öklid ve Minkowski uzaklıkları kullanılmaktadır. KNN algoritması aşağıdaki adımlardan oluşmaktadır [27].

- K komşu değerlerinin belirlenmesi
- Mesafe fonksiyonları yardımıyla uzaklık değerlerinin belirlenmesi
- En yakın olan k komşu değerine sahip olan veri sınıfının belirlenmesi

2.1.2. Destek vektör makineleri (Support vector machines, SVM)

SVM'de her bir sınıfı modellemek yerine, sınıfları birbirinden ayıran iki veya çok boyutlu bir çizgi / eğri bulunur. Bu yapıya hiper düzlem adı verilir. En uygun hiper düzlem, iki sınıf vektörleri arasında en büyük uzaklığa sahip olan karar fonksiyonudur [28, 29].

Kurulan model başarımı hiper düzlem sayesinde tahmin edilebilir. Veriler hiper düzlemde uzaklaştıkça model daha başarılı bir şekilde kurulmuş demektir. İki veya çok boyutlu hiper düzlemde sınıflandırma yapmak için kullanılır [24, 25]. Model eğitildikten sonra tahmin aşaması çok hızlıdır ve yüksek boyutlu verilerle iyi çalışır. Çekirdek yöntemleriyle entegrasyonları, SVM'i çok yönlü hale getirir ve pek çok veri türüne uyum sağlarlar [21].

2.1.3. Naive bayes sınıflandırıcısı (Naive bayes classifier)

Naive Bayes modelleri, yüksek boyutlu veri kümeleri üzerinde hızlı bir şekilde sınıflandırma yapabilen algoritmalarından oluşan bir gruptur. Çok hızlı oldukları ve çok az sayıda ayarlanabilir parametreye sahip oldukları için sınıflandırma problemlerinde tercih edilirler. Naive Bayes sınıflandırıcıları, Bayes sınıflandırma yöntemleri üzerine kuruludur. Bunlar, istatistiksel niceliklerin koşullu olasılıklarının ilişkisini açıklayan bir denklem olan Bayes teoremine dayanır [21, 30]. Naive Bayes algoritmasında etiketli olan verilerin birbirinden bağımsız olduğu varsayımı kabul edilerek tahmin yapılmaktadır.

2.1.4. Karar ağacı (Decision tree, DT)

DT, parametrik olmayan ve bir dizi sorunun sorulduğu sezgisel bir yöntemdir. Sınıflandırma ve regresyon problemlerinin çözümü için kullanılır. Bu algoritma, bir hedef değişkenin değerini tahmin eden modeli oluştururken veri özelliklerinden çıkararak öğrendiği basit karar kurallarını kullanır. Ağaç dallarına koşul ifadeleri belirtilir ve iyi yapılandırılmış bir ağaçta, her soru seçeneklerin sayısını yaklaşık yarı yarıya azaltır, çok sayıda sınıf arasında bile seçenekleri çok hızlı bir şekilde daraltır [21]. Giriş değerinin sağladığı koşula göre dallanma işlemi gerçekleştirilerek sistem çıktısı elde edilir. Veri görselleştirmesi yardımıyla DT algoritmaları oldukça sade ve anlaşılır bir biçimde kullanıcıya cevap verebilirler. Sınıflandırma ağacı oluştururken gini indeksi veya entropi kullanılır, bu iki yöntem düğüm saflığına sınıflandırma hata oranından daha duyarlıdır [24].

2.1.5. Rastgele orman (Random forest, RF)

RF algoritması, pek çok özelliğe ve kayda sahip verilerde tahmine dayalı modeller için başarılı sonuçlar veren bir denetimli öğrenme algoritmasıdır. Bu algoritmada, karar ağaçları rastgele seçilen veri örnekleri üzerinde oluşturulur ve her ağaçtan tahmin alınır. Son olarak oylama yoluyla en iyi çözüm seçilir. Birden fazla karar ağacı yapısını kullanarak basit bir karar ağacından daha doğru tahminler üretmektedir. Hangi öngörücülerin önemli olduğunu otomatik olarak belirleme ve öngörücüler arasındaki karmaşık ilişkileri keşfetme yeteneğine sahiptir [26].

2.2. Derin Öğrenme Mimarileri (Deep Learning (DL) Based Approaches)

Bakım çalışmalarında zamana bağlı uzun vadeli veriler kullanılacağı için ilgili modelin buna uygun olarak seçilmesi gerekmektedir. Derin Öğrenme modellerinden olan Yinelemeli Sinir Ağı yapısının bir dalı olan Uzun Kısa Süreli Bellek ve Geçitli Tekrarlayan Birim uzun vadeli zaman indeksine bağlı olan çalışmalarda başarılı bir öğrenme yapısına sahiptirler [31].

2.2.1. Uzun kısa süreli bellek (Long short-term memory, LSTM)

LSTM, verilerin sıralı olduğu çok çeşitli uygulama alanlarında sıklıkla kullanılan ve başarılı sonuçlara sahip DL mimarilerindedir [31]. LSTM, Yinelemeli Sinir Ağı algoritmalarının karşılaştığı en büyük sorun olan kaybolan gradyan problemlerini, içerisinde bulundurduğu kapılar sayesinde çözüme yeteneğine sahiptir [32, 33]. Burada ana fikir, bellek görevini gören hücre durum vektörü ve

belleğe bilgi giriş çıkışını sağlayan kapılardan oluşan bir blok yapısıdır. Yapı içerisindeki bir birim, bir giriş dizisi üzerinde çalışır ve bir birim içindeki her kapı, tetiklenip tetiklenmediklerini kontrol etmek için sigmoid etkinleştirme işlevini kullanır, durum değişikliğini ve birim içinden akan bilgi eklemesini koşullu hale getirir. LSTM yapısında 3 adet kapı ve 1 adet hücre bulunmaktadır [27, 34].

- **Unutma Kapısı (Forget Gate):** Unutma kapısı, gelen bilgilere göre kendi içerisindeki karar mekanizması sayesinde hangisinin geçip hangisinin unutulacağına karar vermektedir. Unutma kapısı matematiksel olarak Eş. 7'deki gibi ifade edilebilir.
- **Giriş Kapısı (Input Gate):** Giriş kapısı, hücre kapısına gelen durumu güncelleyen kapıdır. Giriş kapısının matematiksel ifadesi Eş. 8'de belirtilmiştir.
- **Çıkış Kapısı (Output Gate):** Çıkış kapısı gizli katmanları güncelleyen kapıdır. Bir sonraki katmana bir önceki katmana ait bilgiyi taşır. Çıkış kapısının matematiksel ifadesi Eş. 9'da belirtilmiştir.
- **Hücre Yapısı (Cell State):** Belirtilen kapıları içerisinde barındıran hücre yapısı sayesinde LSTM algoritmaları kaybolan gradyan problemlerini çözmektedir. Bilginin taşınmasını sağlar. LSTM hücre yapısının matematiksel ifadesi Eş. 10'da belirtilmiştir [35].

$\tilde{c}(t)$ aday durum ve $h(t)$ gizli duruma ait matematiksel ifadeler Eş. 11 ve 12'de görülmektedir.

$$f(t) = \sigma(U_f x(t) + W_f h(t-1) + b_f) \quad (7)$$

$$i(t) = \sigma(U_i x(t) + W_i h(t-1) + b_i) \quad (8)$$

$$o(t) = \sigma(U_o x(t) + W_o h(t-1) + b_o) \quad (9)$$

$$c(t) = f(t) \odot c(t-1) + i(t) \odot \tilde{c}(t) \quad (10)$$

$$\tilde{c}(t) = \tanh(U_c x(t) + W_c h(t-1) + b_c) \quad (11)$$

$$h(t) = o(t) \odot \tanh(c(t)) \quad (12)$$

Eşitliklerde $x(t)$ özellik matrisidir. U_f, U_i, U_c, U_o ve W_f, W_i, W_c, W_o ağırlık matrisleridir. b_f, b_i, b_c, b_o yanlılık vektörleridir. Bu matrisler zamana bağlı değildir, yani bir zaman adımından diğerine değişmezler. Hücre durumu $c(t)$, mevcut zaman adımının hafızasıdır ve giriş kapısı ve unut kapısı sonuçlarına göre güncellenir. Bu durumda iki noktasal çarpma vardır. Birincisi, önceki bilgilerin düşüp düşmeyeceğine karar vermek için unutma vektörü ile önceki bellek arasındadır. Diğeri ise giriş kapısı ile yeni değerlerin eklenmesi için aday değer $\tilde{c}(t)$ arasındadır. Mevcut hücre durumunu güncellemek için her çarpmanın sonuçları eklenir. Gizli durum, filtrelenmiş çıkış hücresi durumuna dayalı olarak bir sonraki diziye taşınacak bilgiyi tutar. Mevcut bloğun çıkışında tanh aktivasyon fonksiyonu ile hücre durumunun değeri -1 ile 1 arasında değişmektedir.

2.2.2. Geçitli tekrarlayan birim (Gated recurrent unit)

GRU yapıları da LSTM yapıları gibi kaybolan gradyan problemlerini çözmek için kullanılan tekniklerden birisidir [36]. GRU algoritmaları yapı olarak LSTM algoritmalarına oldukça benzemektedirler. Aralarındaki temel fark, tek bir geçitleme biriminin, unutma faktörünü ve durum birimini güncelleme kararını eşzamanlı olarak kontrol etmesidir [22]. Yapılarında iki adet kapı bulundurmaktadırlar [27, 37]. Bu kapılar aşağıda belirtilmiştir;

- **Sıfırlama Kapısı (Reset Gate):** GRU yapısında bulunan sıfırlama kapısı sayesinde unutulacak bilgilerin kontrolü yapılmaktadır. Sıfırlama kapısının matematiksel ifadesi Eş. 13'te belirtilmiştir.

- **Güncelleme Kapısı (Update Gate):** Güncelleme kapısı, daha önceden bulunan bilgilerin kaç tanesinin bir sonraki adıma aktarılacağına karar veren yapıdır. Güncelleme kapısının matematiksel ifadesi Eş. 14'te belirtilmiştir.

$\tilde{h}(t)$ aday gizli durum ve $h(t)$ gizli duruma ait matematiksel ifadeler Eş. 15 ve 16'da görülmektedir.

$$r(t) = \sigma(U_r x(t) + W_r h(t-1) + b_r) \quad (13)$$

$$z(t) = \sigma(U_z x(t) + W_z h(t-1) + b_z) \quad (14)$$

$$\tilde{h}(t) = \tanh(x_t U_h + W_t (h(t-1) \odot r_t) + b_h) \quad (15)$$

$$h(t) = (1 - z(t)) \odot h(t-1) + z(t) \odot \tilde{h}(t) \quad (16)$$

U_r, U_z ve U_h mevcut girdinin ağırlık matrisleridir. W_r, W_z ve W_t gizli birimlerin ağırlıklarıdır. b_r, b_z ve b_h yanlılık vektörleridir. Sigmoid işlevi, giriş değerlerini 0 ile 1 arasında dönüştürmek için hem sıfırlama hem de güncelleme kapılarında kullanılır. Sıfırlama kapısı ve gizli durum güncelleme mekanizması, aday duruma entegre edilmiştir. Bu, şu anki yeni bilgidir. Aday gizli durumdaki tanh işlevi, değerlerin -1 ile 1 arasında kalmasını sağlamaktadır.

3. Gerçekleme ve Başarım Sonuçları (Implementation and Achieved Results)

3.1. Veri Seti (Dataset)

Çalışma kapsamında University of California Irvine (UCI) Makine Öğrenmesi Ambarları'ndan alınan UCI Öngörücü Bakım veri seti kullanılmıştır. Veri setinde 14 özellikle tanımlanmış 10.000 örnek bulunmaktadır. Veri setinde yer alan sütunlar aşağıdaki gibidir [38, 39].

Ürün ID (Product ID): Ürün ID olarak belirtilen sütun içerisindeki veriler, kalite etiketleri ve seri numaraları ile verilmektedir. Kalite etiketleri Tip sütununu açıklarken belirtilmiştir.

Tip (Type): Kalite değerleri veri setinde düşük, orta ve yüksek olarak belirtilmiştir. Kısaltma olarak bu değerler sırasıyla L, M, H olarak etiketlenmiştir. Veri setinde düşük kalite değeri %60, orta kalite %29,97 ve yüksek kalite değeri %10,03 olarak verilmiştir.

Hava Sıcaklığı (Air Temperature): Verilerin alındığı sistemdeki hava sıcaklığının Kelvin cinsinden değeridir. Veri setinde hava sıcaklığı en düşük 295,3 K ve en yüksek 304,5 K değerini almaktadır.

İşlem Sıcaklığı (Process Temperature): Verilerin alındığı sistemdeki işlem sıcaklığı karşılık gelmektedir. Kelvin cinsinden ifade edilmiştir. Veri setinde işlem sıcaklığı en düşük 305,7 K ve en yüksek 313,8 K değerini almaktadır.

Dönme Hızı (Rotational Speed): Sistemin devir sayısıdır. Veri setinde bu değerlerin ortalaması 1538,77 rpm ve standart sapması ise 179,28 olarak verilmiştir.

Tork (Torque): Sistemin tork değeridir. Birimi Newton metre (Nm) olarak verilmiştir. Veri setinde bu değerlerin ortalaması 40 Nm ve en yüksek değeri ise 76.6 Nm olarak verilmiştir.

Takım Aşınması (Tool Wear): Sistemdeki malzeme aşınma değeridir. Birimi dakika olarak verilmiştir. Veri setinde en düşük 0 ve en yüksek 253 dakika olarak verilmiştir.

Makine Arızası (Machine Failure): Makine arızası etiketi sistemin herhangi bir yerinde hata uyarısı geldiğinde 1 değerini alan etiket değeridir. Hangi hata türünün geldiği bu etiket için fark etmemektedir.

Hata Tipi (Failure Type): İlgili işlemin vermiş olduğu hata tiplerini barındıran etiket değeridir. Veri setindeki toplam hata tipi sayısı 5'tir [40]. Bu hatalar ve veri setindeki sayısı Tablo 1'de verilmiştir.

Tablo 1. Hata tipleri ve sayıları (Failure types and numbers)

Hata Tipi	Sayısı
Hata Yok (No Failure, NF)	9652
Isı Dağılımı Hatası (Heat Dissipation Failure, HDF)	115
Güç Hatası (Power Failure, PWF)	91
Aşırı Zorlama Hatası (Overstrain Failure, OSF)	78
Takım Aşınma Hatası (Tool Wear Failure, TWF)	46
Rastgele Hata (Random Failure, RF)	18

Hata tiplerinin dağılım grafiği Şekil 2' de gösterilmektedir. Hata tiplerinin dağılımına bakıldığında %33,05 oranı ile HDF'nin en yüksek ve %5 ile RF'nin en düşük orana sahip olduğu görülmektedir. Veri setindeki hatalı kayıtların oranı ile hatasız kayıtların oranı karşılaştırıldığında dengesiz bir dağılım mevcuttur. Bu nedenle deneyler öncesinde veri setinde Sentetik Azınlık Aşırı Örnekleme Yöntemi (Synthetic Minority Over-Sampling Technique, SMOTE) ile veri seti dengeleme işlemi yapılmıştır [41, 42].

- HDF: Bu hata iki koşulun sağlanmasıyla ortaya çıkar. İşlem sıcaklığı ve hava sıcaklığı arasındaki farkın 8.6 Kelvin değerinin altında olması ve dönme hızının 1380 rpm değerinden daha düşük olması durumudur.
- PWF: Bu hata sistem gücünün 3500W değerinden küçük olması veya 9000W değerinden büyük olması durumunda üretilir. Sistemin gücü tork ve dönme hızına bağlı olarak değişmektedir.
- OSF: Bu hata sistem gereksinimlerinin aşırı zorlandığı anlarda oluşmaktadır. Tip ve takım aşınması değerlerine bağlıdır. Bu hatanın alınabileceği koşullar aşağıda belirtilmiştir. Parça kalitesi arttıkça aşırı zorlanma değerleri artmaktadır.
- (Tip = L) & (Takım Aşınması x Tork > 11.000)
- (Tip = M) & (Takım Aşınması x Tork > 12.000)
- (Tip = H) & (Takım Aşınması x Tork > 13.000)
- TWF: Bu hata sistemdeki malzeme aşınma değerinin 200-240 dakika arasında olduğu durumlarda alınmaktadır. Veri setinde bu aralıktaki 120 adet veri bulunmaktadır. Bu değerler, belirtilen zaman aralığında 46 kez başarısız, 74 kez takım değişimi gerektirecek şekilde rastgele atanır.
- RF: Bu hata hiçbir değişkene bağlı olmayıp rastgele gözlemlenen hata değerleridir.

3.2. Başarım Analizi (Performance Analysis)

Başarım incelemeleri yapılırken makine öğrenmesi modellerinde beş katlı çapraz doğrulama yapılmıştır [43]. Başarım metrikleri olarak doğruluk, kesinlik, geri çağırma ve F1-Skor kullanılmıştır. Karışıklık matrisi tahmin edilen ve gerçek sınıflandırmaların değerlerini gösteren bir matristir. Tablo 2'de 2x2 boyutlu bir karışıklık matrisi görülmektedir [44].

Tablo 2. Karışıklık matrisi (Confusion matrix)

↓ Gerçek / Tahmin Edilen→	Pozitif	Negatif
Pozitif	TP (Gerçek Pozitif)	FP (Yanlış Pozitif)
Negatif	FN (Yanlış Negatif)	TN (Gerçek Negatif)

Doğruluk, Model tarafından bir veri seti üzerinden yapılan doğru ve yanlış tahminlerin oranıdır ve Eş. 17'de görüldüğü şekilde hesaplanmaktadır.

$$\text{Doğruluk} = \frac{TP+TN}{TP+FP+FN+TN} \quad (17)$$

Kesinlik, Gerçek Pozitifler ile tüm Pozitifler arasındaki oranı ifade etmektedir.

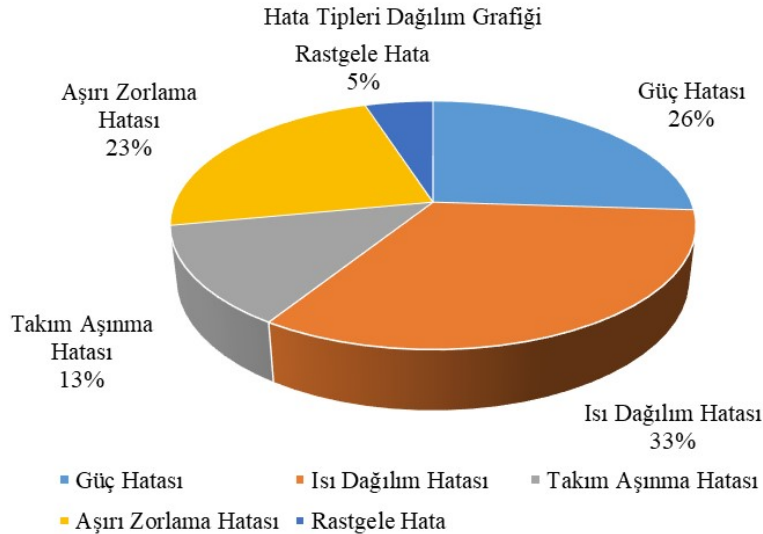
$$\text{Kesinlik} = \frac{TP}{TP+FP} \quad (18)$$

Geri çağırma, Gerçek Pozitif tahminlerin doğru bir şekilde tanımlanmasının ölçüsüdür.

$$\text{Geri Çağırma} = \frac{TP}{TP+FN} \quad (19)$$

F1-Skor, kesinlik ve geri çağırma değerlerinin harmonik ortalamasıdır. Sınıflandırıcı performansının ölçümünde kullanılmaktadır.

$$F1 - \text{Skor} = \frac{2}{\frac{1}{\text{Kesinlik}} + \frac{1}{\text{Geri Çağırma}}} \quad (20)$$

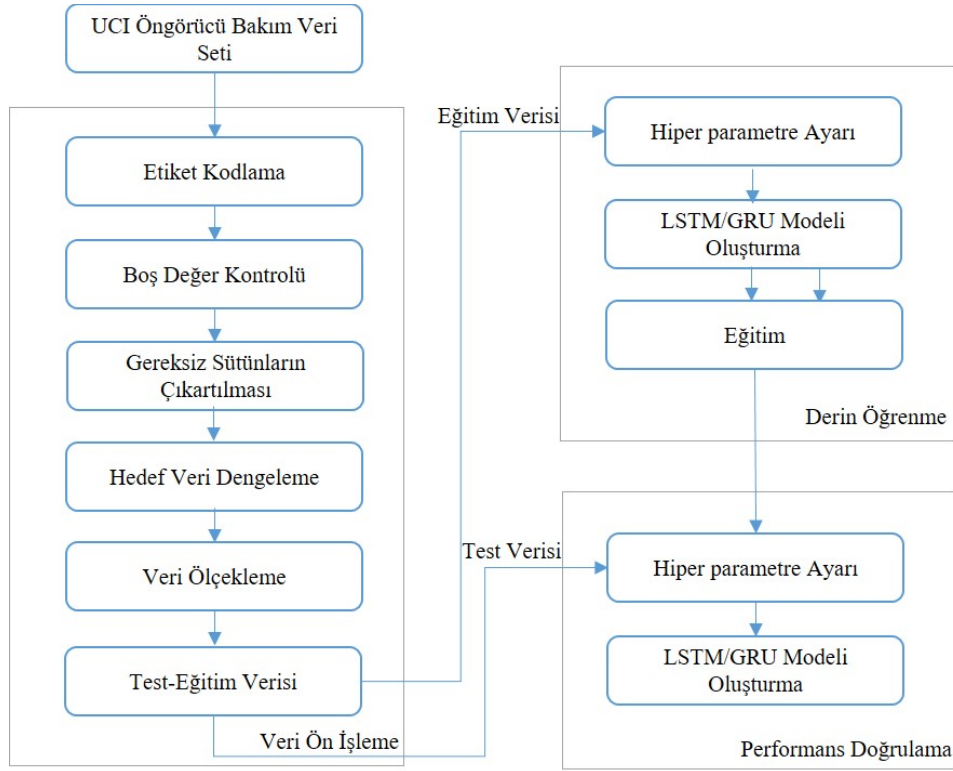
**Şekil 2.** Hata tipi dağılım grafiği (Error type distribution chart)

3.2.1. Klasik makine öğrenmesi temelli yöntem (Classical machine learning based method)

Klasik ML yöntemlerinin veri seti üzerindeki başarımlarını incelemeleri Şekil 3'te verilen akış diyagramı takip edilerek gerçekleştirilmiştir. KNN algoritmasında kullanılan k-komşu değeri 5 olarak alınmıştır. SVM algoritmasında ise çekirdek fonksiyonu olarak 'Radial Basis Function' ve ceza parametresi olarak ise 5 seçilmiştir. Karar Ağacı olarak CART (Sınıflandırma ve Regresyon Ağacı) algoritması kullanılmıştır. Karar Ağacı, Lojistik Regresyon, Rastgele Orman ve Naive Bayes Sınıflandırıcı için çalışma parametre değerleri olarak kütüphanedeki varsayılan değerler kullanılmıştır. Yöntemlerin başarımlarını sonuçları Tablo 3'te verilmiştir. Yöntemlerin başarımlarını

detaylarını gözlemek için Şekil 4'te görülen karışıklık matrisleri oluşturulmuştur.

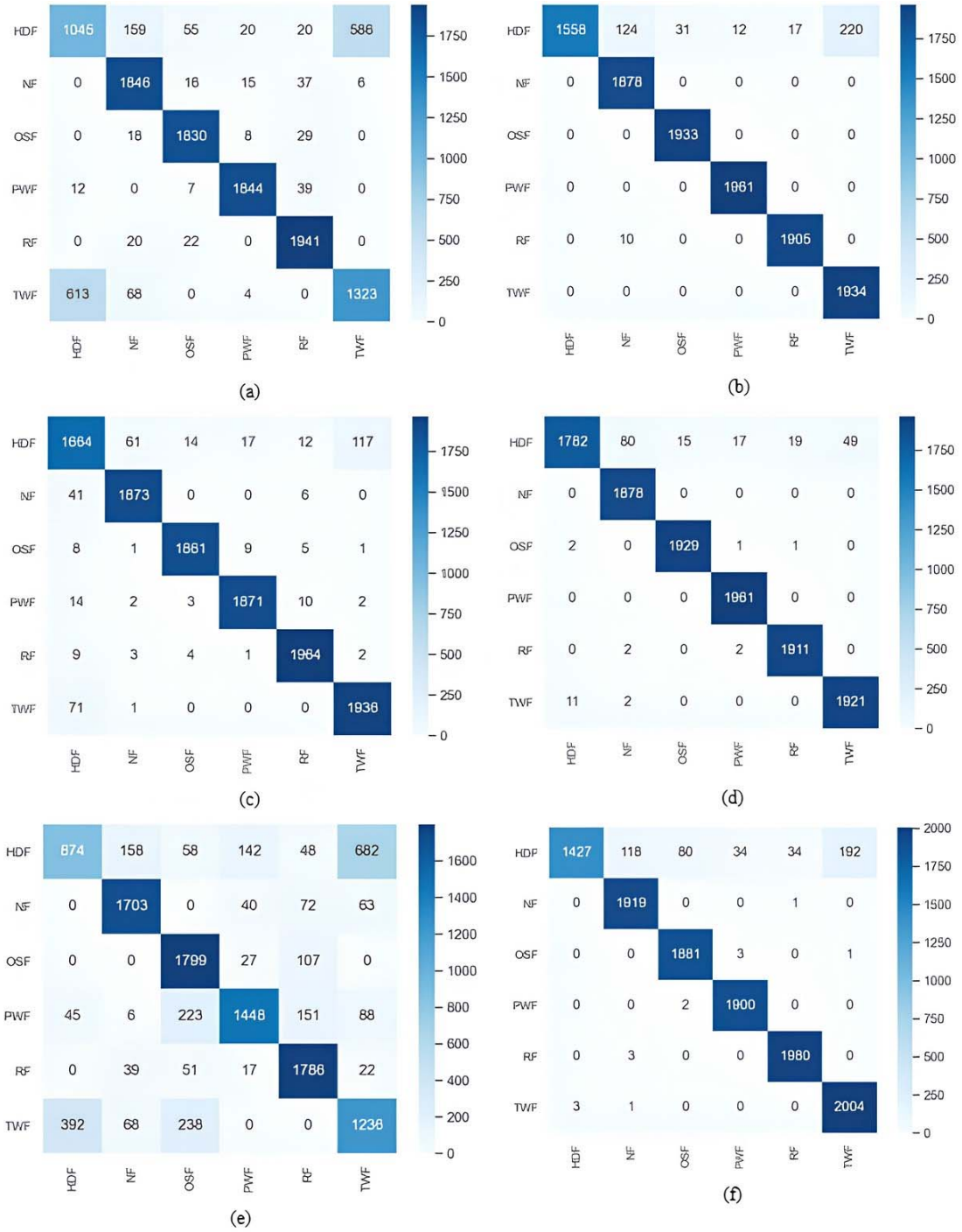
UCI öngörücü bakım veri seti ile ilgili literatürde yer alan çalışmalar genellikle ikili sınıflandırma problemi biçiminde ele alınmıştır. Bu nedenle bu çalışmada kullanılan makine öğrenmesi yöntemlerinin başarımlarını literatür ile karşılaştırmak amacıyla problem aynı zamanda ikili sınıflandırma problemi olarak ele alınmış ve sonuçlar Tablo 4'te sunulmuştur. Literatürde yer alan ikili sınıflandırma çalışmalarında veri dengeleme işlemi yapılmadan başarımlarını sonuçları verilmiştir. Bu nedenle bu çalışmada da ikili sınıflandırma probleminin çözümü için önerilen modellerin başarımlarını incelemesi yapılırken veri dengeleme kullanılmamıştır.



Şekil 3. Akış diyagramı (Flow chart)

Tablo 3. 5 Katlı için çoklu sınıflandırma başarımlarını sonuçları (Multiclass classification performance results)

Yöntem	Doğruluk	Detay Metrikler	HDF	NF	OSF	PWF	RF	TWF
LR	84,86	Kesinlik	55,44	96,17	97,09	96,95	97,88	65,89
		Geri Çağırma	62,58	87,45	94,82	97,52	93,95	69,09
		F1-Skor	58,79	91,59	95,94	97,23	95,88	67,45
SVM	95,81	Kesinlik	79,41	100	100	100	95,12	100
		Geri Çağırma	100	93,34	98,42	99,39	91,98	89,79
		F1-Skor	88,52	96,56	99,20	99,69	93,52	94,62
DT	96,43	Kesinlik	88,28	97,55	98,73	98,37	99,04	96,41
		Geri Çağırma	92,09	96,50	98,84	98,58	98,35	94,07
		F1-Skor	90,14	97,02	98,81	98,47	98,69	95,23
RF	98,26	Kesinlik	90,83	100	99,79	100	99,79	99,33
		Geri Çağırma	99,28	95,72	99,23	98,99	98,96	97,51
		F1-Skor	94,86	97,81	99,51	99,49	99,38	98,41
GNB	76,37	Kesinlik	44,55	90,68	93,07	73,84	93,26	63,91
		Geri Çağırma	66,67	86,27	75,94	86,50	82,53	59,11
		F1-Skor	53,41	88,42	83,64	79,67	87,57	61,42
KNN	95,90	Kesinlik	75,70	99,95	99,79	99,89	99,85	99,80
		Geri Çağırma	99,79	94,02	95,82	98,09	98,26	91,22
		F1-Skor	86,09	96,89	97,77	98,98	99,05	95,32



Şekil 4. Makine öğrenmesi yöntemlerinin karışıklık matrisleri (Confusion matrices of machine learning methods)
(a) LR (b) SVM (c) DT (d) RF (e) GNB (f) KNN

3.2.2. Derin öğrenme temelli yöntem (Deep learning based method)

Çalışma kapsamında değerlendirilen DL algoritmalarının veri seti üzerindeki başarımlarının hesaplanması için Şekil 5'te görülen akış takip edilmiştir.

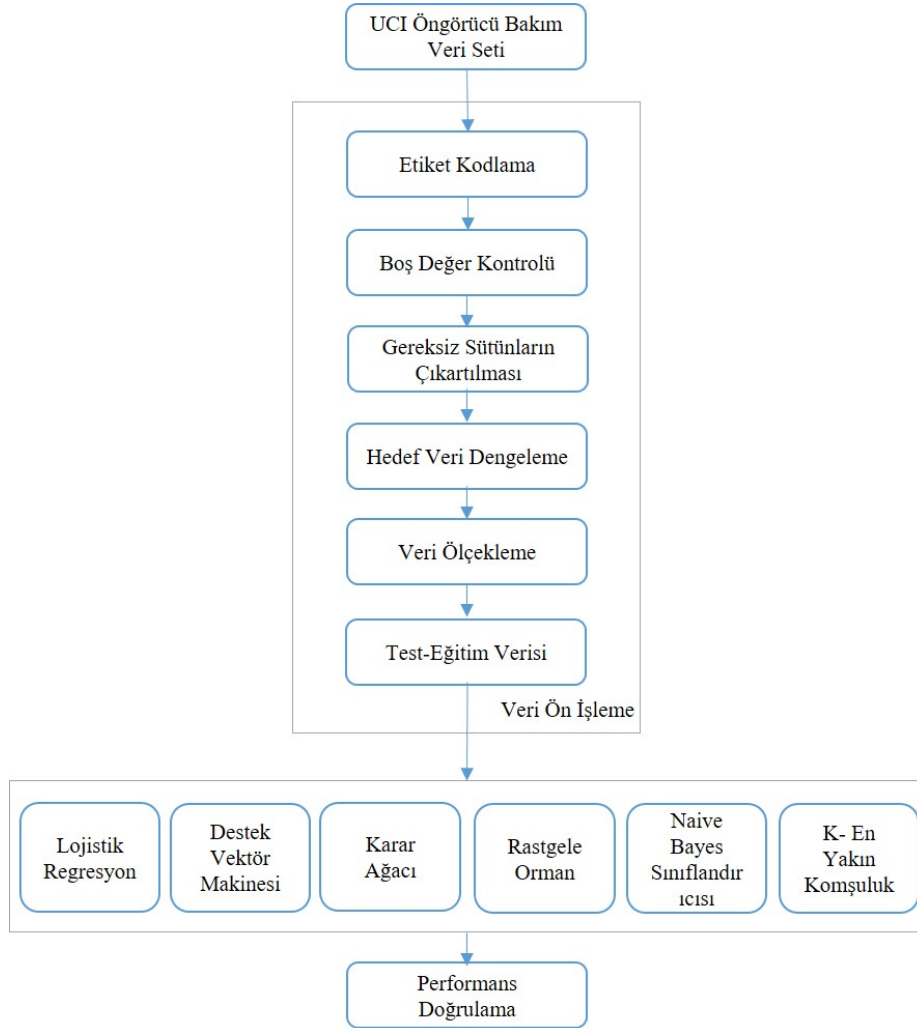
GRU ve LSTM algoritmaları için model kurma aşamasında her biri 200 birim içeren üç katman kullanılmıştır. Kayıp fonksiyonu 'Kategorik Çapraz Entropi', optimizasyon fonksiyonu 'Adam',

seyreltme oranı ise 0,1 olarak belirlenmiştir. Model için eğitim turu sayısı 100 ve küme boyutu 50 alınarak işlemler gerçekleştirilmiştir.

Yapılan tahmin modeli için hata tiplerinin metrik değerleri Tablo 5'te belirtilmiştir. Tablo incelendiğinde OSF, RF ve PWF etiketine sahip hata tiplerinin oldukça başarılı bir şekilde tahmin edildiği gözlemlenmektedir. Mimarilerin başarımlarını detaylarını gözlemlemek için Şekil 6'da görülen karışıklık matrisleri oluşturulmuştur.

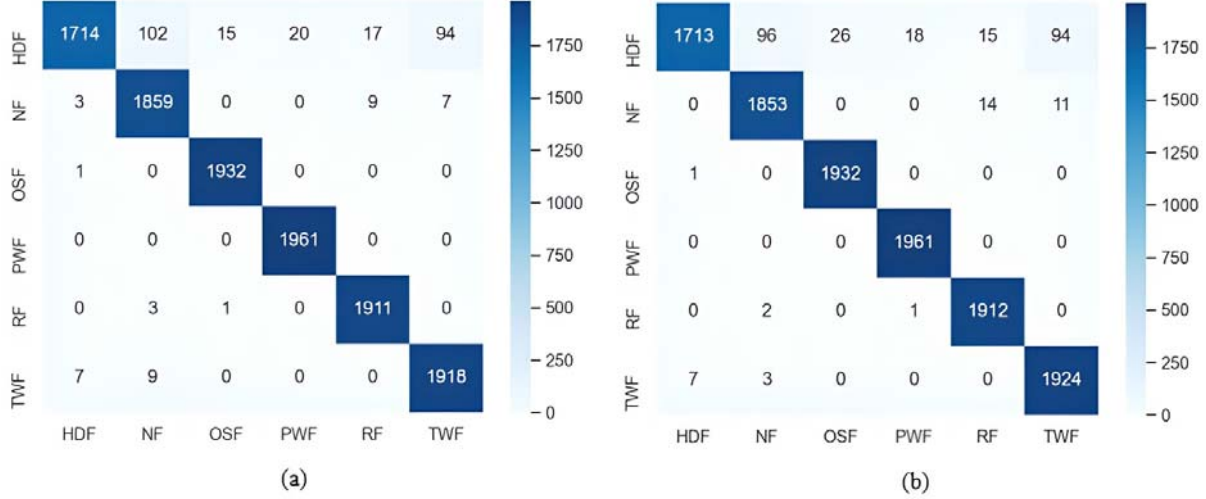
Tablo 4. İkili sınıflandırma başarımları sonuçları (Binary classification performance results)

Yöntem	Doğruluk	Detay Metrikler	F	NF
LR	96,60	Kesinlik	99,69	15,07
		Geri Çağırma	96,87	64,71
		F1-Skor	98,26	24,44
SVM	96,90	Kesinlik	99,48	20,00
		Geri Çağırma	97,37	56,52
		F1-Skor	98,42	29,55
DT	96,85	Kesinlik	98,34	49,32
		Geri Çağırma	98,57	50,68
		F1-Skor	98,14	57,81
RF	97,15	Kesinlik	99,64	23,08
		Geri Çağırma	97,47	68,18
		F1-Skor	98,54	34,48
GNB	96,55	Kesinlik	98,91	26,15
		Geri Çağırma	97,55	44,74
		F1-Skor	98,23	33,01
KNN	96,65	Kesinlik	99,69	16,44
		Geri Çağırma	96,92	66,67
		F1-Skor	98,29	26,37

**Şekil 5.** Derin öğrenme akış diyagramı (Deep learning flow chart)

Tablo 5. Derin öğrenme mimarileri başarımları sonuçları (Deep learning performance results)

Yöntem	Doğruluk	Detay Metrikler	HDF	NF	OSF	PWF	RF	TWF
GRU	97,51	Kesinlik	87,36	98,99	99,95	100	99,79	99,17
		Geri Çağırma	99,36	94,22	99,18	98,99	98,66	95,00
		F1-Skor	92,98	96,55	99,56	99,49	99,22	97,04
LSTM	97,51	Kesinlik	87,31	98,67	99,95	100	99,84	99,48
		Geri Çağırma	99,54	94,83	98,67	99,04	98,51	94,83
		F1-Skor	93,02	96,71	99,31	99,18	99,17	97,10


Şekil 6. Derin öğrenme modellerinin karışıklık matrisleri (Confusion matrices of deep learning models) (a) GRU (b) LSTM

Tablo 6. Çok sınıflı sınıflandırma karşılaştırma tablosu (Multiclass classification comparison table)

Yöntem	Doğruluk (Dengelenmiş Veri)	Referans
LR	84,86	Bu çalışmada
SVM	95,81	*
DT	96,43	*
RF	98,26	*
GNB	76,37	*
KNN	95,90	*
GRU	97,51	*
LSTM	97,51	*
Dengelenmiş K-Star	98,75	[45]

Ayrıca, literatürde yer alan çalışmalar ile bu çalışmada kullanılan çok sınıflı sınıflandırma için tüm yöntemlerin doğruluk metriği bazındaki başarımların karşılaştırması Tablo 6'da verilmiştir.

Çoklu sınıflandırma sonuçları değerlendirildiğinde klasik makine öğrenmesi yöntemlerinden RF'in %98,26 ile en yüksek başarımları verdiği görülmektedir. Bu başarımların Dengelenmiş K-Star ile ulaşılan başarımların değeri olan %98,75'e oldukça yakındır, aralarında %0,49'lük bir fark bulunmaktadır. Derin öğrenme mimarileriyle ise %97,51'lik doğruluk değeri elde edilmiştir.

UCI öngörücü bakım veri seti ile ilgili literatürde yer alan çalışmalar genellikle ikili sınıflandırma problemi biçiminde ele alınmıştır. Bu nedenle bu çalışmada kullanılan derin öğrenme modellerinin başarımlarını literatür ile karşılaştırmak amacıyla problem aynı zamanda ikili sınıflandırma problemi olarak ele alınmış ve sonuçlar Tablo 7'de sunulmuştur. GRU ve LSTM algoritmaları için model kurma aşamasında her biri 200 birim içeren üç katman kullanılmıştır. Kayıp fonksiyonu 'İkili Çapraz Entropi', optimizasyon fonksiyonu 'Adam', seyreltme oranı ise 0,1 olarak belirlenmiştir. Model için

eğitim turu sayısı 100 ve küme boyutu 10 alınarak işlemler gerçekleştirilmiştir.

Tablo 7. İkili sınıflandırma başarımları sonuçları (Binary classification performance results)

Yöntem	Doğruluk	Detay Metrikler	F	NF
GRU	97,00	Kesinlik	99,38	26,15
		Geri Çağırma	97,56	58,62
		F1-Skor	98,46	36,17
LSTM	97,05	Kesinlik	99,28	30,77
		Geri Çağırma	97,71	58,82
		F1-Skor	98,49	40,40

Ayrıca, literatürde yer alan yöntemler ile bu çalışmada kullanılan tüm yöntemlerin doğruluk metriği bazındaki başarımların karşılaştırması Tablo 8'de verilmiştir.

Açık literatürde yer alan çalışmalar arasında ikili sınıflandırma için en yüksek doğruluk değerinin veri dengeleme yapılmadan %98,34 olduğu görülmektedir. Karşılaştırma amacıyla ikili sınıflandırma yöntemlerinin başarımları veri seti üzerinde veri dengelemesi

Tablo 8. İkili sınıflandırma karşılaştırma tablosu (Binary classification comparison table)

Yöntem	Doğruluk (Orijinal)	Doğruluk (Dengelemiş Veri)	Referans
LR	96,60	80,86	Bu çalışmada
SVM	96,90	93,70	*
DT	96,85	92,02	*
RF	97,15	95,03	*
GNB	96,55	82,65	*
KNN	96,65	92,59	*
GRU	97,00	93,03	*
LSTM	97,05	92,20	*
Torbalama Ağaçları Topluluk Sınıflandırıcısı	98,34	-	[39]
LR ve RF	96,02 ve 97,22	-	[40]
İkili Lojistik Regresyon	97,10	-	[46]
Hibrit Denetimli ve Denetimsiz Makine Öğrenmesi	98,46	-	[47]
Uzman Kurallarla Gelişen Bulanık Sinirsel Sınıflandırıcı (EFNC-Exp)	97,30	-	[48]
Kendi Kendine Organize Yön Tabanlı Veri Bölümleme (SODA)	96,80	-	[48]

yapılmadan da incelenmiştir. Veri dengelemesi yapılmadan (orijinal veri ile) makine öğrenmesi yöntemleri arasında en yüksek doğruluk değeri RF ile %97,15 olarak gerçekleşmiştir. Derin öğrenme mimarileri arasında ise en yüksek doğruluk değeri LSTM ile %97,05 olarak elde edilmiştir. Veri dengeleme sonrasında ise makine öğrenmesi yöntemleri arasında en yüksek doğruluk değeri yine RF ile %95,03 olarak sağlanmıştır. Derin öğrenme mimarilerinde GRU en yüksek doğruluk değerine %93,03 olarak ulaşmıştır. Literatürde yer alan başarımların değerleri veri dengelemesi yapılmadan orijinal veri için elde edilmiştir. Orijinal veri için ulaşılan en yüksek doğruluk değeri %98,46 ile Hibrit Denetimli ve Denetimsiz Makine Öğrenmesi yöntemi kullanılarak elde edilmiştir. Bu çalışmada orijinal veri için elde edilen yüksek doğruluk değerleri RF için %97,15 ve LSTM için %97,05'dir. Bu değerler literatürde elde edilen en yüksek değerlerden %1,31 ve %1,41 daha düşüktür.

4. Sonuçlar (Conclusions)

Bu çalışmada, UCI Makine Öğrenmesi Ambarlarından alınan öngörücü bakım veri seti üzerinde hata tipi tespiti çoklu sınıflandırma problemi olarak ele alınmıştır. Problemin çözümünde klasik makine öğrenmesi yöntemleri ve derin öğrenme mimarilerinin başarımları karşılaştırmalı olarak incelenmiştir. Yöntemlerin başarımlarını incelemeleri veri seti üzerinde veri dengeleme işlemi yapıldıktan sonra 5 katlı çapraz doğrulama kullanılarak gerçekleştirilmiş ve başarımların sonuçları geri çağırma, F1-skor, kesinlik ve doğruluk metrikleri kullanılarak sunulmuştur. Makine öğrenmesi yöntemleri arasında en yüksek doğruluk değeri RF ile %98,26 olarak elde edilmiştir. Derin öğrenme mimarilerinin her ikisi için doğruluk değeri %97,51 olarak gerçekleşmiştir.

Ayrıca, açık literatürde yer alan ve aynı veri setini kullanan çalışmalarla başarımların karşılaştırması yapabilmek için problem ikili sınıflandırma problemi olarak da ele alınmıştır. Bu durum için ise makine öğrenmesi yöntemleri arasında en yüksek doğruluk değeri %97,15 olarak yine RF ile elde edilmiştir. Derin öğrenme mimarileri arasında en yüksek doğruluk değeri LSTM ile %97,05 olarak gerçekleşmiştir.

Yapılan çalışmalar sonucunda bu veri seti üzerinde klasik makine öğrenmesi teknikleri ve derin öğrenme tekniklerinin birbirine yakın sonuçlar verdiği gözlemlenmiştir. Bu nedenle, bu problemin çözümünde klasik makine öğrenmesi yöntemleri ile sınıflandırma

yapmanın zaman ve maliyet açısından daha uygun olduğu değerlendirilmiştir.

Gelecek çalışmalar kapsamında frekans bölgesi özellikleri ve optimizasyon teknikleri kullanılarak başarımın iyileştirilmesi hedeflenmektedir. Ayrıca, bu makalede kullanılan makine öğrenmesi yöntemleri ile derin öğrenme mimarilerinin farklı öngörücü bakım veri setleri üzerindeki başarımları karşılaştırmalı olarak incelenecektir.

Teşekkür (Acknowledgement)

Yazarlar, UCI Makine Öğrenmesi Ambarlarına teşekkür ederler, [<http://archive.ics.uci.edu/ml>].

Kaynaklar (References)

- Dangut M. D., Skaf Z., Jennions I.K., An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset, ISA Transactions, 113, 127-139, 2021.
- Mobley R. K., An introduction to predictive maintenance, Second Edition, Elsevier Science, USA, 2002.
- De Faria H., Costa J. G. S., Olivias J. L. M., A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis, Renew. Sustain. Energy Rev., 46, 201-209, 2015.
- Yu T., Zhu C., Chang Q., Wang J., Imperfect corrective maintenance scheduling for energy efficient manufacturing systems through online task allocation method, Journal of Manufacturing Systems, 53, 282-290, 2019.
- Wang Y., Deng C., Wu J., Wang Y., Xiong Y., A corrective maintenance scheme for engineering equipment, Engineering Failure Analysis, 36, 269-283, 2014.
- Ran Y., Zhou X., Lin P., Wen Y., Deng R., A Survey of Predictive Maintenance: Systems, Purposes and Approaches, ArXiv, 1-36, 2019.
- Sharma D. K., Brahmachari S., Singhal K., Gupta D., Data driven predictive maintenance applications for industrial systems with temporal convolutional networks, Computers & Industrial Engineering, 169, 2022.
- Garcia M. C., Sanz-Bobi M. A., del Pico J., SIMAP: Intelligent System for Predictive Maintenance. Application to the health condition monitoring of a windturbine gearbox, Comput. Ind., 57 (6), 552-568, 2006.
- Hashemian H. M., Bean W. C., State-of-the-art predictive maintenance techniques, IEEE Trans. Instrum. Meas., 60 (10), 3480-3492, 2011.

10. Liao L., Köttig F., Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction, *IEEE Transactions on Reliability*, 63, 191-207, 2014.
11. Türe B. A., Akbulut A., Zaim A.H., Techniques for apply predictive maintenance and remaining useful life: A systematic mapping study, *BSEU Journal of Science*, 8, 497-511, 2021.
12. Hermawan A. P., Kim D. S., Lee J. M., Predictive maintenance of aircraft engine using deep learning technique, 2020 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island - Korea, 1296-1298, 21-23 Ekim, 2020.
13. Louridas, P., Ebert, C., *Machine learning*, IEEE Software, 33 (5), 110–115, 2016.
14. Jones, P. J., Catt M., Davies M. J., Edwardson C. L., Mirkes E. M., Khunti K., Yates T., Rowlands A. V, Review Feature selection for unsupervised machine learning of accelerometer data physical activity clusters – A systematic review, 90, 120-128, 2021.
15. Ouadah A., Zemmouchi-Ghomari Salhi L., N., Selecting an appropriate supervised machine learning algorithm for predictive maintenance, *The International Journal of Advanced Manufacturing Technology*, 4277-4301, 2022.
16. Mueller A. C., Guido S., *Introduction to Machine Learning with Python*, O'Reilly Media, United States of America, 2016.
17. Güemes- Peña D., Nozal C. L., Sánchez R. M., Maudes J., Emerging topics in mining software repositories: Machine learning in software repositories and datasets, *Progress in Artificial Intelligence*, 7 (5), 237-247, 2018.
18. Raschka S., *Python Machine Learning*, Packt Publishing, Birmingham, UK, 2016.
19. Rovira M., Engvall K., Duwig C., Identifying key features in reactive flows: A tutorial on combining dimensionality reduction, unsupervised clustering, and feature correlation, *Chemical Engineering Journal*, 438, 1-15, 2022.
20. Ahfock D., McLachlan G.J., Semi-supervised learning of classifiers from a statistical perspective: A brief review, *Econometrics and Statistics*, 1-25, 2022.
21. VanderPlas J., *Python Data Science Handbook Essential Tools for Working with Data*, O'Reilly Media, United States of America, 2017.
22. B. Yoshua, Goodfellow I. J., Courville A., *Deep Learning*, MIT Press, 2016.
23. Sutton R. S., Barto A. G., *Reinforcement Learning, An Introduction* second edition, The MIT Press Cambridge, Massachusetts London, England, 2018.
24. James G., Witten D., Hastie Tibshirani T., R., *An Introduction to Statistical Learning with Applications in R*, Springer, 2021.
25. Zhang S., Zhang S., Wang B., Habetler T.G., *Deep Learning Algorithms for Bearing Fault Diagnostics – A Comprehensive Review*, IEEE Access, 8, 29857-29881, 2020.
26. Bruce P., Bruce A., Gedeck P., *Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python*, O'Reilly Media, United States of America, 2020.
27. S.Raschka, V.Mirjalili, *Python Machine Learning*, Packt Publishing, 2018.
28. Cortes, C., Vapnik, V., Support vector networks. *Machine Learning*, 20, 273–297, 1995.
29. Gökdemir A., Çalhan A., Deep learning and machine learning based anomaly detection in internet of things environments, *Journal of the Faculty of Engineering and Architecture of Gazi University* 37 (4), 1945-1956, 2022.
30. Ross S.M., *Introduction to Probability Models*, Introduction to Probability Theory, Elsevier, 2014.
31. Sherstinsky A., *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network*, *Physica D: Nonlinear Phenomena*, 404 (8), 132306, 2020.
32. Hochreiter S., Schmidhuber J., Long Short-Term Memory, *Neural Computation*, 9 (8), 1735-1780, 1997.
33. Akın M., Sağıroğlu Ş., Short term traffic speed prediction with RNN method for roads characterized by density-based clustering method, *Journal of the Faculty of Engineering and Architecture of Gazi University* 37 (2), 581-593, 2022.
34. Dangut M.D., Skaf Z., Jennions I.K. Rescaled-LSTM for predicting aircraft component replacement under imbalanced dataset constraint, 2020 Advances in Science and Engineering Technology International Conferences (ASET), Dubai-United Arab Emirates, 1-9, 02-04 Şubat, 2020.
35. Kong Z., Cui Y., Xia Z., Lv H., Convolution and long short-term memory hybrid deep neural networks for remaining useful life prognostics, *Appl. Sci.*, 9 (19), 2019.
36. Dey R., Salem F.M., Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks, 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, Massachusetts-USA, 1597-1600, 6-9 Ağustos, 2017.
37. Czum J. M., *Dive Into Deep Learning*, J. Am. Coll. Radiol., 17 (5), 637–638, 2020.
38. Hagemeyer S., Mauthe F., Zeiler P., Creation of publicly available data sets for prognostics and diagnostics addressing data scenarios relevant to industrial applications, *International Journal of Prognostics and Health Management*, 12 (2), 2153-2648, 2021.
39. Matzka S., Explainable artificial intelligence for predictive maintenance applications, 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), Irvine, CA-USA, 69-74, 21-23 Eylül, 2020.
40. Katkar R., Buktar R., Big data and predictive analytics in manufacturing enterprises for enhanced decision making, *Journal of Emerging Technologies and Innovative Research (JETIR)*, 8 (9), 374-384, 2021.
41. Chawla N. V., Bowyer K. W., Hall L. O., Kegelmeyer W. P., SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*, 16, 321–357, 2002.
42. Elreedy D., Atiya A. F., A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance, *Information Sciences*, 505 (2019), 32-64, 2019.
43. Yılmaz E., Fetal State Assessment from Cardiotocogram Data Using Artificial Neural Networks, *J. Med. Biol. Eng.*, 36, 820–832, 2016.
44. Kohavi, R., & Provost, F. (1998). Glossary of terms. *Machine Learning*, 30 (2–3), 271–274.
45. Ghasemkhani, B., Aktas, O., Birant D., Balanced K-Star: An Explainable Machine Learning Method for Internet-of-Things-Enabled Predictive Maintenance in Manufacturing, *Machines*, 11 (3), 322, 2023.
46. Iantovics L. Bl & Enăchescu C., Method for Data Quality Assessment of Synthetic Industrial Data, *Sensors* 2022, 22, 1608, 2022.
47. Harichandran A., Raphael B., Mukherjee A., Equipment activity recognition and early fault detection in automated construction through a hybrid machine learning framework, *Computer-Aided Civil and Infrastructure Engineering*, 1-16, 2022.
48. Souza, P.V.C.; Lughofer, E. EFNC-Exp: An evolving fuzzy neural classifier integrating expert rules and uncertainty. *Fuzzy Sets Syst.*, in press, 2022.