Researcher

CILT/VOL **02**
SAYI/ISSUE **01**
YIL/YEAR **2022**

ANKARA BİLİM ÜNİVERSİTESİ

Researcher

CİLT/VOL **02** SAYI/ISSUE **01** YIL/YEAR **2022**

e-ISSN:2717-9494

Yayıncı / Publisher: Ankara Bilim Üniversitesi / Ankara Science University
Basım Tarihi / Date of Publication: Temmuz 2022/ July 2022
Yayın Türü / Publication Type: Uluslararası Süreli Yayın / International Periodical
İletişim Bilgileri / Contact Information: Çamlıca Mah.Anadolu Bulvarı No:16A/1 Yenimahalle Ankara
Web Sitesi / Website: https://researcher.ankarabilim.edu.tr/
E-posta / E-mail: researcher@ankarabilim.edu.tr

Researcher uluslararası, hakemli ve yılda iki sayı yayınlanan dergidir. İngilizce ve Türkçe dilindeki metinler kabul edilir.
The Researcher is a peer-reviewed, international journal publishing two issues a year. Manuscripts are accepted in English and Turkish languages.

ANKARA, TEMMUZ 2022 / JULY 2022

Önsöz

Yayın hayatına 2013 yılında başlamış olan "Researcher: Social Sciences Studies" (RSSS), 2020 Ağustos ayı itibariyle "Researcher" ismiyle Ankara Bilim Üniversitesi bünyesinde yayın hayatına devam etmektedir. Fen Bilimleri alanına katkıda bulunmayı hedefleyen özgün araştırma makalelerinin yayımlandığı bir dergidir. Dergi, özel sayılar dışında yılda iki kez yayımlanmaktadır.

Amaçları doğrultusunda dergimizin yayın odağında; Endüstri Mühendisliği, Bilgisayar Mühendisliği ve Elektrik Elektronik Mühendisliği alanları bulunmaktadır. Dergide yayımlanmak üzere gönderilen aday makaleler Türkçe ve İngilizce dillerinde yazılabilir. Dergiye gönderilen makalelerin daha önce başka bir dergide yayımlanmamış veya yayımlanmak üzere başka bir dergiye gönderilmemiş olması gerekmektedir. Bir makalenin dergide yayımlanabilmesi için en az iki hakem tarafından olumlu rapor verilmesi gerekir.

Değerlendirme sonucu kabul edilen çalışmalar sırasıyla; intihal kontrolünün yapılması, kaynakça düzenlemesi, gönderme ve atıf kontrolü, mizanpaj ve dizgisinin yapılması süreçlerinden geçer.

Researcher, Dergipark üzerinden bilimsel araştırmaların içeriğine anında açık erişim sağlamaktadır.

Researcher makale işlem ücreti (gönderme, değerlendirme veya basım ücreti) ve makalelere erişim için abonelik ücreti talep etmediği için ücretsiz yayın yapan dergi statüsündedir, Dergimiz herhangi bir kâr amacı gütmemekte ve hiçbir gelir kaynağı bulunmamaktadır.

Baş Editör

Doç. Dr. Hakan ÇAĞLAR

Editör

Dr. Öğr. Üyesi Yavuz Selim ÖZDEMİR

İçindekiler / Index

Sıhhiye Bölgesi Hava Kalitesi İndeksinin Aşırı Öğrenme Makineleri ve Yapay Sinir Ağları ile Tahmini

Burhan BARAN..... 1-18

Learning Capabilities of AI Methodologies on Multi-Class Datasets

Ender SEVİNÇ..... 19-28

Reverberation Effect on Online Hazardous Sound Event Detection

Yüksel ARSLAN..... 29-39

Big Data Reduction and Visualization Using the K-Means Algorithm

Hakan AKYOL, Hale Sema KIZILDUMAN, Tansel DÖKEROĞLU..... 40-45

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 2, No. 1, July 2022
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi

Sıhhiye Bölgesi Hava Kalitesi İndeksinin Aşırı Öğrenme Makineleri ve Yapay Sinir Ağları ile Tahmini

Burhan BARAN[1*]
[1]İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Malatya, Türkiye;
ORCID: 0000-0001-6394-412X *Corresponding Author: burhanbaran@gmail.com

Received: 16 February 2022; Accepted: 29 March 2022

Reference/Atıf: B. Baran, "Sıhhiye Bölgesi Hava Kalitesi İndeksinin Aşırı Öğrenme Makineleri ve Yapay Sinir Ağları ile Tahmini", Researcher, vol. 02, no. 01, pp. 1-18, Jul. 2022

Özet

Bu çalışma ile Sıhhiye bölgesindeki hava kalitesi indeksinin (HKİ) hem aşırı öğrenme makineleri (AÖM) hem de yapay sinir ağları (YSA) algoritmaları ile tahmin edilmesi amaçlanmıştır. Bu amaçla, HKİ'yi etkileyebilecek yedi adet parametre seçilmiştir. Bu parametreler PM_{10}, SO_2, CO, sıcaklık, nem, basınç ve rüzgâr hızıdır. İlk olarak, HKİ ile bu yedi parametre arasında korelasyon analizi yapılmıştır. Analiz sonucuna göre HKİ ile en güçlü ilişkinin atmosferik parametrelerden PM_{10} ile, meteorolojik parametrelerden ise basınç ile olduğu sonucuna ulaşılmıştır. 2018 yılının Ağustos, Ekim, Kasım ve Aralık aylarına ait parametre değerleri eğitim verisi olarak belirlenmiştir. 2019 yılının Ocak ve Şubat aylarına ait ilk 14 günlük parametre verileri ise test verisi olarak belirlenmiştir. HKİ değerleri 1 ile 6 arasında matematiksel olarak sınıflandırılmıştır. Sınıflandırma çalışmaları hem ham veriler hem de normalize edilmiş veriler ile gerçekleştirilmiştir. Sınıflandırma sürecinde algoritmalarda farklı eğitim fonksiyonları ve gizli nöron sayıları kullanılmıştır. Sonuçların güvenilirliği için 3-kat çapraz doğrulama yapılmıştır. En yüksek performansa sahip aktivasyon fonksiyonları ve nöron sayıları gerçek test verilerine uygulanmıştır. Son olarak, HKİ'nin matematiksel sınıflandırma sonuçları ile tahmini sınıflandırma sonuçları karşılaştırılmıştır. Elde edilen sonuçlara göre hem ham hem de normalize veriler ile yapılan sınıflandırma çalışmalarında AÖM algoritmasının YSA algoritmasından daha başarılı sonuçlar elde ettiği görülmüştür. Başarım oranları ham verilerde %85.71, normalize verilerde %71.43 olarak gerçekleşmiştir.

Anahtar Kelimeler: hava kalitesi indeksi, aşırı öğrenme makineleri, yapay sinir ağları, sınıflandırma, korelasyon.

Prediction of Air Quality Index of Sihhiye Region by Extreme Learning Machines and Artificial Neural Networks

Abstract

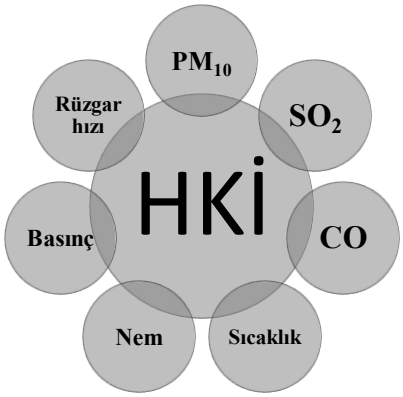
With this study, it was aimed to estimate the air quality index (AQI) in the Sihhiye region with both extreme learning machines (ELM) and artificial neural networks (ANN) algorithms. For this purpose, seven parameters that could affect the AQI had been chosen. These parameters were PM_{10}, SO_2, CO, temperature, humidity, pressure and wind speed. Firstly, correlation analysis was performed between the AQI and these seven parameters. According to the results of the analysis, it was concluded that the strongest relation with the AQI were with PM_{10} from the atmospheric parameters and the pressure from the meteorological parameters. The parameter values for August, October, November and December of 2018 year were determined as training data. The parameter values for the first 14 days of January and February of 2019 year were determined as test data. AQI values were classified mathematically between 1 and 6. Classification studies were applied to both raw data and normalized data. In the classification process, different training functions and hidden neuron numbers were used in algorithms. 3-fold cross-validation was performed for the reliability of the results. The activation function and neuron numbers with highest performance were applied to actual test data. Finally, mathematical classification results were compared with the predicted classification values of AQI. According to the results obtained, in the classification studies conducted with both raw and normalized data, it was observed that ELM algorithm achieved more successful results than ANN algorithm. The success rates were 85.71% in raw data and 71.43% in normalized data.

Keywords: air quality index, extreme learning machine, artificial neural networks, classification, correlation.

1. Giriş

Dünyanın hızla gelişmesine bağlı olarak kentleşme kaçınılmaz hale gelmiştir. Ancak, bu hem ekosistemler hem de insanlar için bir tehlike oluşturmaktadır. İnsanlar için oluşan tehlikelerden biri hava kirliliğidir [1]. Solunan kirli hava sağlığımızı doğrudan etkilemektedir. Bu nedenle, havanın kalitesini optimum seviyede tutmak büyük önem taşımaktadır [2]. Kentleşme ile birlikte, hava kirliliğinin bir diğer sebebi de sanayileşmedir. Özellikle hava kalitesi önlemlerinin bulunmadığı veya asgari düzeyde olduğu endüstriyel bölgelerde hava kalitesinin insan sağlığı üzerinde olumsuz etkisi bulunmaktadır. Bu nedenle, hava kirliliğinin derecesini objektif olarak değerlendirilmesi ve kirletici konsantrasyonlarını doğru şekilde tahmin edilmesi önem arz etmektedir. Bunun için bilimsel hava kalitesi izleme ve erken uyarı sistemleri oluşturulmalıdır [3]. Atmosferdeki hava kalitesi genel olarak hava kalitesi indeksi (HKİ) olarak bilinen bir parametre ile ölçülür. HKİ, havanın kirlilik derecesi hakkında bilgi verir. Hava kirliliği ise havadaki gazlar ve katı parçacıkların bir karışımı olarak tanımlanabilir. Araçlardan çıkan egzoz gazı, fabrikalardan çıkan kimyasallar, katı yakıt kaynaklı partiküller ve tozlar hava kirliliğine neden olan temel faktörlerdir. Atmosferde HKİ değerini etkileyen bazı kirleticiler partikül madde (PM_{10}), kükürt dioksit (SO_2) ve karbon monoksit (CO)'tir. PM_{10}, aerodinamik çapı 10 μm'den küçük partikül madde olarak tanımlanmaktadır. Hem dış hem de iç mekanlarda bulunabilir. Akciğer hastalıkları, kalp damar hastalıkları ve kalp krizine neden olabilmektedir [4]. Birçok sabit ve hareketli kaynaktan oluşabilmektedir. Küçük boyutlu olanların akciğerler üzerinde olumsuz etkileri olabilmektedir. SO_2, kömür ve fuel-oil gibi yakıtların yanması sonucu oluşmaktadır. İnsanların solunum fonksiyonlarını etkiler. Sülfürik asit oluşumuna ve sülfür dioksit birikmesine katkıda bulunmaktadır. Hem PM_{10} hem de SO_2, özellikle kış aylarında kentsel hava kalitesi sorunları ile yakından ilişkilidir [5]. CO ise büyük miktarlarda solunduğunda zararlı olabilen bir gazdır. Renksiz ve kokusuzdur. Zehirlenmeye bağlı olarak bayılma ve ölümlere neden olabilmektedir. Dış havaya salınan en büyük CO kaynakları taşıtlardır [6].

Hava kirliliği insan sağlığını etkileyecek seviyelerin üzerine çıktığında hava kalitesi tahmin teknikleri geliştirilmektedir. Hava kalitesi tahmininde genellikle geleneksel yaklaşımlar, matematiksel ve istatistiksel teknikler kullanılmaktadır. Geleneksel tahmin modelleri temel olarak bilgisayar altyapısı gerektirmektedir. Çalışmalar HKİ'yi tahmin etmek için yeni modelleme yaklaşımlarını ortaya koymuştur. AÖM ve YSA ile sınıflandırma da bu yaklaşımlar kapsamındadır [7]. Ayrıca, hava kirleticilerinin dağılımı için rüzgâr hızı, sıcaklık, basınç, nem gibi meteorolojik faktörler de önemlidir. Bu parametrelerin hava kirliliği ve insan sağlığı açısından herhangi bir olumsuzluğu bulunmamaktadır. Ancak PM_{10}, SO_2 ve CO değerlerine yapacakları meteorolojik etkiler ile HKİ değerinin değişmesine neden olabilmektedirler. HKİ değerini etkileyen tüm bu parametrelerin temsili gösterimi Şekil 1'deki gibidir. Birçok ülke hava kirliliğini azaltmak için çalışmalar yapmaktadır. Bu ülkelerden biri de Türkiye'dir. Bu amaçla, ülke genelinde Çevre ve Şehircilik Bakanlığı tarafından kurulmuş Hava Kalitesi Ölçüm İstasyonları bulunmaktadır. Bu istasyonlar aracılığı ile kirleticiler ait veriler toplanmakta ve yine bu istasyonlarda analiz edilmektedir [8]. Analiz sonucunda istasyonun bulunduğu bölgenin hava kirliliği kalitesini gösteren HKİ değerleri hesaplanmaktadır. Çevre ve Şehircilik Bakanlığı Ulusal Hava Kalite İzleme Ağı web sitesinde kullanılan HKİ sınır değerleri şu şekildedir. İyi (0-50), orta (50-100), hassas (100-150), sağlıksız (150-200), kötü (200-300) ve tehlikeli (300-500).



Şekil 1: HKİ Değerini Etkileyen Parametrelerin Temsili Gösterimi

HKİ ve HKİ'yi etkileyebilecek parametrelerin tahminine yönelik makine öğrenmesi ve farklı algoritmalar kullanılarak çok sayıda çalışma yapılmıştır.

Sevinç (2022) tarafından yapılan çalışmada bir karar ağacı tahmincisi ve yeni bir parametre ayarlama süreci ile uyarlanabilir bir yükseltme algoritması kullanarak hastaların ciddiyetini tahmin etmek için geliştirilmiş bir öğrenme modeli önerilmiştir [9]. Cihan vd. (2021) tarafından yapılan çalışmada bir sanayi bölgesindeki PM_{10} ve $PM_{2.5}$ bileşenlerinin tahmini için uyarlamalı ağ tabanlı bulanık çıkarım sistemi, destek vektör regresyonu, sınıflandırma ve regresyon ağaçları, rastgele orman, k-en yakın komşuluk ve aşırı öğrenme makine yöntemleri kullanılmıştır. ANFIS modeli, diğer yöntemlere kıyasla PM_{10} değerlerini tahmin etmede daha başarılı olmuştur [10]. Baran (2021) tarafından Beşiktaş'taki hava kalitesi indeksinin yapay sinir ağları ve k-en yakın komşuluk (kNN) algoritmaları ile tahmin edilmesi üzerine bir çalışma yapılmıştır [11]. Shishegaran vd. (2020) günlük HKİ tahmini için bir çalışma yapmışlardır. Dört tahmin modeli kullanılmışlardır. Modelleri karşılaştırmak için ise maksimum negatif ve pozitif hatalar, ortalama kesirli sapma, mutlak yüzde hata, kök ortalama kare hatası ve normalleştirilmiş kare hata yöntemlerini kullanılmışlardır. Elde edilen sonuca göre doğrusal olmayan topluluk modeli en iyi performansı göstermiştir [12]. Wang vd., (2020) yaptıkları çalışma ile günlük hava kalitesi tahmini için yenilikçi bir hibrit model önermişlerdir. Çalışmalarında aykırı nokta tespiti ve düzeltme algoritmasını benimsemişlerdir. Tahmin etkinliğini değerlendirmek için ise hipotez testi kullanılmıştır. Önerdikleri hibrit model diğer modellere göre daha yüksek tahmin seviyesine ulaşmıştır [13]. Baran (2019) tarafından yapılan çalışmada rüzgâr hızının ve buna bağlı olarak rüzgârdan elde edilebilecek enerjinin aşırı öğrenme makineleri algoritması tarafından tahmini yapılmıştır [14]. Liu vd. (2019) tarafından yapılan çalışma ile Pekin kenti için HKİ, İtalya için ise NO_X parametresinin tahmini amaçlanmıştır. Bu tahminler için destek vektör regresyonunu ve rastgele orman regresyonunu kullanılmışlardır. Regresyon modellerinin performansını değerlendirmek için ortalama karekök hatası ve korelasyon katsayısı kullanılmıştır. Yapılan deneysel çalışmalar neticesinde rastgele orman regresyonu modelin daha iyi performans gösterdiği, destek vektör regresyon modelin ise çok sayıda verinin işlenmesi yöntemlerine uygun olmadığı sonucuna ulaşılmıştır [15]. Sevinç (2019) tarafından yapılan çalışmada tek gizli katmanlı ileri beslemeli sinir ağları ile entegre yeni bir evrimsel öznitelik seçim algoritması önerilmiştir. Önerilen algoritma genetik algoritmaların evrimsel tekniğini birleştirmektedir. Aşırı öğrenme makineleri ile seçilen her bir özellik alt kümesinin uygunluk değerini hesaplanmıştır [16]. Zou vd. (2019), geçmişe ait hava kalitesi ve meteorolojik verilerin bir derin sinir ağı olan airQP-DNN modelinde kullanılmak suretiyle gelecekteki HKİ değerlerinin tahmini yapmışlardır. Bu tahmini HKİ verilerine göre de açık hava etkinlikleri için rota planlaması yapılmıştır. Çalışmada Pekin ve çevre kentlere ait bir yıllık veri seti kullanılmıştır. Elde ettikleri deneysel sonuçlara göre önerilen modelin diğer yaygın olarak kullanılan yöntemlerden daha iyi performans gösterdiği sonucuna ulaşmışlardır [17]. Bai vd. (2016) PM_{10}, SO_2 ve NO_2 içeren günlük hava kirliliği konsantrasyonlarını tahmin etmek için dalgacık tekniği ve geri yayılım sinir ağı modelini kullanarak bir W-BPNN modeli geliştirmişlerdir. Geliştirdikleri modeli, Çin'in Chongqing Nan'an Bölgesinde test etmişlerdir [18]. Baran (2019) tarafından yapılan çalışmada aşırı öğrenme makineleri algoritması ile hava kalitesi indeksinin tahmini gerçekleştirilmiştir [19]. Zhang ve Ding (2017) Hong Kong'un Sham Shui Po ve Tap Mun bölgelerindeki iki izleme istasyonunda altı yıllık veriler de dahil olmak üzere sekiz hava kalitesi parametresinden elde edilen verileri kullanarak AÖM algoritmasını eğitmişlerdir. Eğitilen AÖM algoritmasını kullanarak hava kirleticilerinin tahmini üzerine bir çalışma yürütmüşlerdir [20]. Sevinç (2018) tarafından yapılan çalışmada tek gizli katmanlı ileri beslemeli sinir ağlarında hesaplama parametrelerinin etkinliği incelenmiştir [21]. Zhu vd., (2017) yaptıkları çalışmada HKİ tahmininin doğruluğunu arttırmak için bölgesel hava kalitesi indeksi için bir tahmin modeli geliştirmişlerdir. Bu amaçla, ampirik mod ayrıştırma-EMD-SVR-Hibrit ve EMD-IMFs-Hibrit olarak adlandırılan iki hibrit model önermişlerdir [22].

Peng vd., (2017) doğrusal olmayan makine öğrenme yöntemleri kullanarak Kanada'da $PM_{2.5}$, O_3 ve NO_2 parametrelerine ait yoğunluk tahminleri yapmışlardır. Elde ettikleri verileri kullanmak suretiyle de HKİ değerinin tahminini yapmışlardır [23]. Jose (2017) Madrid kentindeki NO_2 kirliliğini tahmin etmek için olasılıksal tahmin tekniğini geliştirmişlerdir [24]. Patrico ve Ernesto (2016) Santiago kentindeki saatlik $PM_{2.5}$ konsantrasyonunu tahmin etmek için ileri beslemeli bir sinir ağı modeli kullanılmışlardır [25]. Avşar (2015) tarafından yapılan çalışmada Burhaniye İlçesinde SO_2, NO_x, CO, O_3 ve VOC

parametrelerini içeren bir hava kalitesi analizi yapılmıştır [26]. Vong vd. (2014) PM_{10} seviyesinin sınıfını tahmin etmek için AÖM algoritmasını kullanarak bir uyarı sistemi oluşturulmuşlardır. Sonuçlarını destek vektör makineleri algoritmasının sonuçları ile karşılaştırmışlardır. Tahminleri iyileştirmek için önceden çoğaltma adı verilen dengesizlik stratejisini uygulamışlardır [27]. Moustris vd. (2010) Yunanistan'ın Atina Bölgesi'nde 2001 ve 2005 yılları arasındaki eşik değer üzerindeki kirlilik değerlerinden en az birini dikkate almak suretiyle Avrupa Bölgesel Kirlilik Endeksi'nin maksimum günlük değerini tahmin etmek için YSA algoritmasını kullanmışlardır [28]. Biancofiore vd. (2017), PM_{10} konsantrasyonu tahmin etmek için özyinelemeli sinir ağı modeli kullanmışlardır. Kullandıkları model % 95 oranında doğru tahmin yapmıştır [29]. Mekparyup vd. (2020), tarafından yapılan çalışma ile Tayland'ın doğu bölgesinde bulunan sekiz hava izleme istasyonu tarafından ölçülen verilerin multi layer perceptron algoritmasına uygulanması suretiyle HKİ değerinin tahmini amaçlanmıştır. Yapılan analizler neticesinde HKİ tahmininde O_3 ve PM_{10} parametrelerinin önemli rol oynadığı, NO_2, SO_2 ve CO parametrelerinin ise çok daha az önemli olduğu tespit edilmiştir. Ayrıca, HKİ değerinin yaz mevsimi sonunda düşük, yaz aylarında orta, kış aylarında yüksek olduğu tespit edilmiştir. Sınıflandırma tahmininde ise multi layer perceptron algoritması %90 oranında başarım elde etmiştir [30]. Liu vd. (2017) Çin'in Pekin, Tianjin ve Shijiazhuang bölgelerindeki HKİ tahmini için mevcut makine öğrenme algoritmalarının tahmin hatasını en aza indirecek tahmin sonuçlarını elde etmek için destek vektör regresyon algoritmasını kullanmışlardır [31]. Ganesh vd. (2017) HKİ değerini tahmin etmek amacıyla YSA kullanmışlardır. YSA algoritmasını eğitmek için ise birçok yöntem kullanmışlardır. Bunlardan bazıları Elman sinir ağı, radyal temel fonksiyon sinir ağı, çok katmanlı algılayıcıdır. Çalışmalarında Houston ve Los Angeles bölgelerinde 2010-2016 yılları için NO_2, CO, O_3, $PM_{2.5}$, SO_2 ve PM_{10} konsantrasyonları bağımsız, HKİ ise bağımlı değişken olarak kullanılmıştır [32]. Saatcioglu vd. (2011) tarafından çalışmada Marmaray Projesi ile otomobil kullanımının azaltılmasının 2015 ve 2030 yılları için İstanbul'daki hava kirliliğini nasıl etkileyeceğini değerlendirilmiştir. Elde edilen sonuçlara göre tüm kirletici türleri için emisyonlardaki azalma oranının 2015 yılında %12.4 ve 2030 yılında %11.6 olacağı sonucuna ulaşılmıştır [33]. Dragomir (2010) tarafından k en yakın komşuluk algoritması ile HKİ değerinin tahminine yönelik bir çalışma yapılmıştır. Algoritmanın giriş parametreleri olarak SO_2, NO_2, O_3 ve CO parametrelerini kullanılmıştır. 2009 yılı Haziran ayının 29 günlük verileri ile bu çalışma yapılmıştır. K en yakın komşuluk algoritması ise Weka yazılımı aracılığı ile uygulanmıştır. K en yakın komşuluk algoritması tarafından 29 günden 19 tanesi doğru tahmin edilmiştir [34]. Jiao vd. (2019) tarafından yapılan çalışma ile HKİ değerinin tahmini amaçlanmıştır. Uzun kısa süreli bellek algoritması kullanılmıştır. $PM_{2.5}$, PM_{10}, SO_2, sıcaklık, rüzgâr yönü, NO_2, CO ve O_3 algoritmada kullanılan parametrelerdir. Çalışma sonucunda Uzun kısa süreli bellek algoritmasının HKİ değerini iyi derecede tahmin edebileceğine dair sonuçlara ulaşılmıştır [35]. Kadılar ve Kadılar (2017) ise mevsimsel otoregresif entegre hareketli ortalama yöntemi kullanarak Aksaray ilindeki hava kalitesini etkileyen SO_2 parametresinin tahmini üzerine bir çalışma yapmışlardır [36]. Avşar vd. (2010) tarafından yapılan çalışmada İstanbul'daki yol süpürme makinelerinden kaynaklanan gürültü seviyesinin ve PM_{10} konsantrasyonunun etkisi incelenmiştir. Çalışma sonucunda bu araçlardan kaynaklanan PM_{10} konsantrasyonunun 355 $\mu g/m_3$'e ulaştığı ve bunun hem yayalara hem de operatöre yüksek oranda etki ettiği sonucuna ulaşılmıştır [37].

Geçmişte Sıhhiye bölgesi için en önemli kirletici kaynaklar araç trafiği ve fosil yakıtlardı. Fosil yakıtlar nedeniyle SO_2 seviyesi yüksekti. Ancak, günümüzde ısıtmada doğalgazın kullanılmasına bağlı olarak SO_2 seviyesi düşmüştür. Bununla birlikte, hızla artan araç sayısı, PM değerinin yüksek çıkmasına sebep olabilmektedir. Bu çalışmada PM_{10}, SO_2, CO, sıcaklık, nem, basınç ve rüzgâr hızı parametreleri dikkate alınarak AÖM ve YSA algoritmaları tarafından HKİ değerlerine ait sınıflandırma tahminlerinin yapılması amaçlanmıştır. Elde edilen tahmini HKİ sınıflandırma sonuçları ile matematiksel HKİ sınıflandırma sonuçları karşılaştırılmıştır. Ayrıca, ham verilerin normalize edilmesi durumunda tahmin sonuçlarının nasıl etkileneceği de incelenmiştir. Tahmin çalışmalarında bağımsız giriş değişkenleri olarak PM_{10}, SO_2, CO, sıcaklık, nem, basınç ve rüzgâr hızı parametreleri kullanılmıştır. Bağımlı çıkış değişkeni ise HKİ'ye ait sınıflandırma değerleri olmuştur. HKİ değerlerini elde etmek için AQI Calculator [38] uygulaması kullanılmıştır. Bu uygulama PM_{10}, $PM_{2.5}$, SO_2, NO_x, CO, O_3 ve NH_3 gibi yedi adet kirleticinin konsantrasyon değerlerine göre HKİ değerini hesaplamaktadır. Bu yedi parametreden en az üçü girilerek HKİ değeri hesaplanabilmektedir. Çalışmada 2018 yılının Ağustos, Ekim, Kasım ve Aralık aylarına ait 123 satırdan oluşan veriler eğitim verisi olarak kullanılmıştır. 2019

yılının Ocak ve Şubat aylarının ilk 14 günlük verileri ise test verisi olarak kullanılmıştır. Hem ham hem de normalize edilmiş veriler kullanılarak ayrı ayrı HKİ sınıflandırma değeri tahminleri yapılmıştır. AÖM ve YSA tarafından 2019 yılına ait bu iki farklı 14 günlük veriye ait HKİ sınıflarının doğru tahmin edilmesi amaçlanmıştır. En yüksek başarım oranında ve en kısa sürede doğru sonuçlar veren eğitim fonksiyonu ve nöron sayısı gerçek test verilerine uygulanmıştır. Elde edilen tahmin sonuçları karşılaştırılarak tahmin yöntemlerinin başarımları ölçülmüştür. Literatürdeki çalışmalar incelendiğinde genellikle PM_{10}, SO_2, CO parametreleri kullanılmışken bu çalışmada ek olarak sıcaklık, nem, basınç ve rüzgâr hızı parametreleri de HKİ değerinin belirlenmesinde giriş parametresi olarak dikkate alınmıştır. Ayrıca ham verilerin yanında normalize veriler ile de çalışmalar yapılmıştır. Bu bağlamda hem meteorolojik hem de atmosferik parametrelerin dikkate alındığı AÖM ve YSA algoritmaları ile yapılan bu tahmini sınıflandırma çalışmasının insan sağlığını olumsuz etkileyebilecek HKİ değerinin tahmin edilmesinde yol gösterici olacağı düşünülmektedir.

Çalışmanın bu aşamadan sonraki kısmı şu aşamalardan oluşmaktadır. İlk olarak HKİ değerini etkileyebilecek giriş parametreleri ile HKİ arasında korelasyon analizi yapılmıştır. Parametrelerin HKİ'yi hangi oranda etkilediğinin tespiti amaçlanmıştır. Ardından, AÖM ve YSA'ya ait teknik bilgiler verilmiştir. Daha sonra durum çalışmalarına yer verilmiştir. Giriş verilerinin hem ham hem de normalize edilmesi durumunda AÖM ve YSA ile elde edilen sonuçlara değinilmiştir. Sonuç bölümünde ise çalışmada elde edilen sonuçların karşılaştırması yapılmıştır.

2. İlgili Çalışmalar

2.1. HKİ Hesaplaması ve Sınır Değerler

HKİ günlük hava kalitesi indeksidir. Havanın kirlilik derecesi hakkında bilgi verir. Bu çalışmada kullanılan HKİ değerleri AQI Calculator uygulaması tarafından hesaplanmıştır. AQI Calculator PM_{10}, $PM_{2.5}$, SO_2, NO_X, CO, O_3 ve NH_3 parametrelerine ait değerlerin en az üç tanesinin girilmesi ile hava kalitesinin değerini hesaplayan excel tabanlı bir uygulamadır. Bu çalışmada PM_{10}, SO_2 ve CO parametrelerine ait değerler AQI Calculator uygulamasına girilerek HKİ değerleri elde edilmiştir. Elde edilen bu HKİ değerleri çalışma boyunca "hesaplanan HKİ" olarak adlandırılmıştır. AQI Calculator uygulaması tarafından elde edilen hesaplanan HKİ değerleri Çevre ve Şehircilik Bakanlığı Ulusal Hava Kalite İzleme Ağı web sitesindeki HKİ değerleri ile doğrulanmıştır [39]. Hava kirliliğinin kalitesini belirlemek için HKİ değerine belirli aralıklarla sınır değerler konulmuştur. Çevre ve Şehircilik Bakanlığı Ulusal Hava Kalite İzleme Ağı web sitesinde kullanılan HKİ sınır değerleri şu şekildedir. 0-50 (iyi), 50-100 (orta), 100-150 (hassas), 150-200 (sağlıksız), 200-300 (kötü) ve 300-500 (tehlikeli) olarak sınıflandırılmıştır. Bu çalışmada kullanılan sınır değerler ve atanan sınıf değerleri ise 0-50 (1. sınıf), 51-100 (2. sınıf), 101-150 (3. sınıf), 151-200 (4. sınıf), 201-300 (5. sınıf), 301-500 (6. sınıf) şeklindedir.

2.2. Eğitim Veri Seti

İlk olarak PM_{10}, SO_2 ve CO parametreleri AQI Calculator uygulamasına girilerek hesaplanan HKİ değerleri elde edilmiştir. Daha sonra, Ankara ili Sıhhiye bölgesine ait 2018 yılı Ağustos, Ekim, Kasım ve Aralık aylarındaki sıcaklık, nem, basınç ve rüzgâr hızı verileri www.timeanddate.com adlı web sitesinden alınmıştır [40]. Böylece, 7 adet giriş verisine karşılık 1 adet çıkış verisi olan hesaplanan HKİ verilerini içeren 123 satırlık eğitim veri seti oluşturulmuştur. Toplamda yedi adet olan meteorolojik ve atmosferik giriş parametresine göre elde edilen 123 adet hesaplanan HKİ değeri 1 ile 6 arasında matematiksel olarak sınıflandırılmıştır. Bu sınıflandırma işlemi sonucunda 1. sınıftan 45 adet, 2. sınıftan 66 adet, 3. sınıftan 8 adet ve 4. sınıftan 4 adet veri elde edilmiştir. Ancak, 5. ve 6. sınıflar için veri üretilmemiştir. Bunun nedeni, hesaplanan HKİ verileri içinde 201 ile 500 değerleri arasında hiçbir HKİ verisinin bulunmamasıdır.

2.3. Korelasyon Analizi

Korelasyon katsayısı (r), iki özellik arasındaki ilişkinin önemini ölçmede kullanılmaktadır. +1 ile -1 arasında değerler almaktadır ve birimi yoktur. r değeri sıfırdan büyük ise, parametreler arasında pozitif bir ilişki olduğu, sıfırdan küçük ise negatif bir ilişki olduğu sonucuna ulaşılmaktadır. Çıkan sonucun sıfıra eşit olması durumu ise, parametreler arasında doğrusal bir ilişki olmadığını göstermektedir. Pearson korelasyon katsayısı eşitlik 1 ile hesaplanmaktadır [41].

$$r = \frac{\sum xy - \frac{\sum x \sum y}{n}}{\sqrt{\left(\sum x^2 - \frac{(\sum x)^2}{n}\right)\left(\sum y^2 - \frac{(\sum y)^2}{n}\right)}} \tag{1}$$

Bu eşitlikte x ve y, iki sürekli yapının özelliklerini göstermektedir.

Korelasyon analizi, 2018 yılının Ağustos, Ekim, Kasım ve Aralık aylarına ait parametre değerleri ile hesaplanan HKİ değerleri karşılaştırılarak gerçekleştirilmiştir. Hesaplanan HKİ ile meteorolojik ve atmosferik parametreler arasında oluşan korelasyon grafikleri ve korelasyon katsayıları Şekil 2'deki gibidir.

2.4. Aşırı Öğrenme Makineleri

AÖM [42] tarafından geliştirilmiştir. İki katmandan oluşmaktadır. İkinci katmanı eğitilebilen ileri beslemeli bir yapay sinir ağıdır [43]. Klasik öğrenme algoritmaları ile karşılaştırıldığında yerel minimum, aşırı uyum gibi problemleri yaşamadan çok daha kısa sürede ve daha iyi performansta sonuçlar elde edilebilmektedir. AÖM, regresyon ve sınıflandırma gibi çalışmalarda kullanılmaktadır. N-gizli düğümü olan tek gizli katmanlı ileri beslemeli yapay sinir ağı eşitlik 2'deki gibi tanımlanmıştır [44].

$$f_N(x) = \sum_{i=1}^{N} B_{i,} G(a_{i,} b_{i,} x), x \in R, a_i \in R) \tag{2}$$

Burada, a_i ve b_i öğrenme parametreleridir. B_i, i. gizli düğümün ağırlığıdır. G(x) ise aktivasyon fonksiyonudur.

2.5. Yapay Sinir Ağları

YSA'lar 1943 yılında Mcculloch and Pitts tarafından geliştirilmişlerdir. Öğrenme yeteneğine sahip örüntü tanıma ve sınıflandırma tekniğidir. Biyolojik sinir ağlarını taklit eden sentetik yapılardır [45, 46, 8]. YSA'larda kullanılan ileri beslemeli geri yayılım sinir ağı algoritmaları, giriş vektörüne bağlı olarak çıkış vektörünü hesaplamada kullanılan algoritmalardır. Hem lineer olmayan hem de karmaşık problemleri çözme başarısından dolayı çoğunlukla tercih edilmektedir [47]. Yapay sinir ağları için ileri beslemeli toplama fonksiyonu ve transfer fonksiyonu sırasıyla eşitlik 3 ve eşitlik 4' te gösterilmiştir [48].

$$I_i = \sum_b W_{bi} x_b \tag{3}$$

$$y_i = f(I_i) \tag{4}$$

Bu çalışmadaki YSA çalışmasında ileri beslemeli geri yayılım sinir ağı algoritması kullanılmıştır. Kullanılan sinir ağı yedi adet giriş verisi, 123 adet gizli nöron ve 1 adet çıkış verisine göre tasarlanmıştır. Bu çalışmada kullanılan algoritmalar Matrix Laboratory (MATLAB) ortamında çalıştırılmıştır.

2.6. Normalizasyon

Yapay sinir ağlarında normalize edilmiş verilerle çalışmak daha hızlı sonuç vermektedir. Bu çalışmada ham verilerin normalize edilmesinde "min-maks normalizasyon" tekniği kullanılmıştır. Tüm eğitim ve test verileri dikkate alınarak 0 ile 1 arasında normalizasyon işlemi gerçekleştirilmiştir. Kullanılan normalizasyon denklemi eşitlik 5'teki gibidir [49].

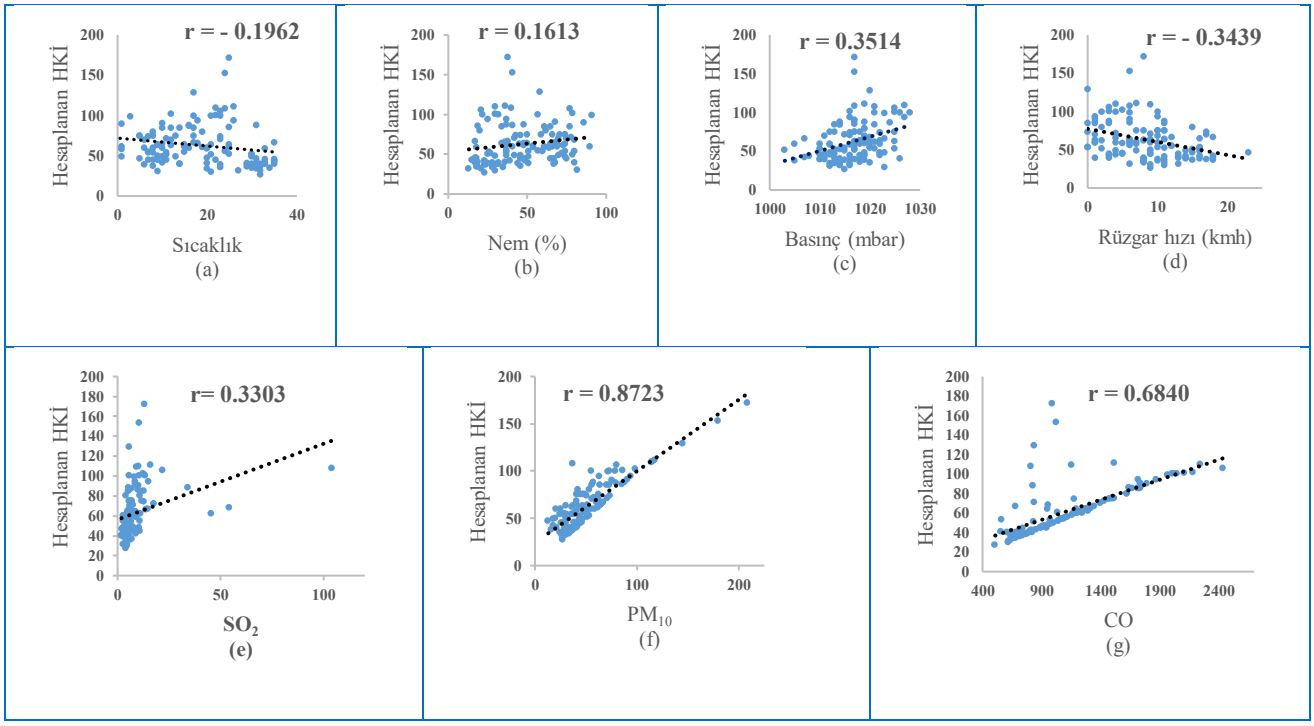
$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{5}$$

Burada, x' normalize edilmiş veriyi gösterirken x_i giriş verisi x_{max} giriş kümesindeki en büyük sayıyı, x_{min} ise giriş kümesindeki en küçük sayıyı göstermektedir.

3. Yapılan Çalışma ve Bulgular

3.1. Parametreler Arası Korelasyon Analizi

Korelasyon analizi, 2018 yılının Ağustos, Ekim, Kasım ve Aralık aylarına ait parametre değerleri ile hesaplanan HKİ değerleri karşılaştırılarak gerçekleştirilmiştir. Hesaplanan HKİ ile meteorolojik ve atmosferik parametreler arasında oluşan korelasyon grafikleri ve korelasyon katsayıları Şekil 2'deki gibidir.



Şekil 2: Hesaplanan HKİ ile Giriş Parametreleri Arasındaki Korelasyon Analizi
a) Sıcaklık b) Nem c) Basınç d) Rüzgâr hızı e) SO_2 f) PM_{10} g) CO

Hesaplanan HKİ ve meteorolojik parametreler arasındaki ilişki incelendiğinde, HKİ ile nem ve basınç arasında pozitif ilişki olduğu görülürken, sıcaklık ve rüzgâr hızı ile arasında negatif ilişki olduğu görülmektedir. Meteorolojik parametreler ile olan en güçlü ilişki basınç ile, en zayıf ilişki ise nem ile olmuştur. Diğer taraftan, hesaplanan HKİ ile atmosferik parametrelerin tamamında pozitif ilişki olduğu görülmektedir. Atmosferik parametreler ile olan en güçlü ilişki PM_{10} ile, en zayıf ilişki ise SO_2 ile olmuştur.

3.2. Ham Veriler ile AÖM Çalışması

AÖM algoritması ile yapılan sınıflandırma çalışmalarda *sinüs, sigmoidal* ve *hardlimit* aktivasyon fonksiyonları kullanılmıştır. Bu aktivasyon fonksiyonlarının test süreleri ve test doğruluk oranları karşılaştırılmıştır. Karşılaştırma sonucunda elde edilen değerler EK bölümündeki Tablo E1'de gösterilmiş-tir. Yapılan birinci karşılaştırmada kullanılan gizli nöron sayısı 50 olarak seçilmiştir. 50 gizli nöron seçilmesinin nedeni, AÖM algoritmasının ilk aşamada nasıl tepki verdiğini görmek ve diğer nöronların sayısını belirlemektir. AÖM algoritmasında, çıkış ağırlıkları analitik olarak hesaplanırken, giriş ağırlıkları rastgele hesaplanmaktadır. Bu nedenle, programın her çalışmasında elde edilen sonuçlar birbirine yakın fakat farklı değerler olabilmektedir. Bu sorunun üstesinden gelmek için AÖM algoritması ile yapılan değerlendirmede 3 kat çapraz doğrulama tekniği kullanılmıştır. Bu çalışmada maksimum 120 gizli nöron kullanılmıştır. Elde edilen sonuçlar test doğruluk oranları ve aktivasyon fonksiyonlarına göre incelendiğinde, en yüksek değerin %76.42 test doğruluk oranı ve 0.000520 saniyelik test süresi ile *hardlimit* fonksiyonu tarafından karşılandığı görülmektedir. Hem *hardlimit* hem de *sine* ve *sigmoid* aktivasyon fonksiyonlarının farklı gizli nöron sayılarında nasıl tepki vereceğini incelemek için, bu üç aktivasyon fonksiyonunun 6 farklı nöron sayısına bağımlılığı incelenmiştir. Kullanılan gizli nöron sa-yıları tüm gizli nöronlara yakın ara değerlerden seçilmiştir. Bu doğrultuda elde edilen sonuçlar Tablo 1'de görülmektedir.

Tablo 1: Farklı Aktivasyon Fonksiyonu ve Nöron Sayısında Test Değerleri (Ham Veri)

Aktivasyon kodu - gizli nöron sayısı	**Test Süresi (sn.)**	**Ortalama Test Doğruluğu (%)**
sig-10	0.0003429	78.86
sin-10	0.0003433	33.33
hardlim-10	0.0004178	78.86
sig-25	0.0003677	79.68
sin-25	0.0003873	31.71
hardlim-25	0.0004554	66.67
sig-50	0.0004672	80.49
sin-50	0.0004997	34.15
hardlim-50	0.0004544	82.12
sig-75	0.0005035	65.85
sin-75	0.0005426	33.33
hardlim-75	0.0004760	85.37
sig-100	0.0005531	78.05
sin-100	0.0006763	37.40
hardlim-100	0.0004890	81.30
sig-120	0.0006389	72.36
sin-120	0.0006230	29.27
hardlim-120	0.0005296	82.93

Tablo 1'den görülebileceği gibi, en yüksek ortalama test doğruluğu oranı %85.37 oranında ve *hardlim-75* ile gerçekleşmiştir. Sonraki en yüksek test doğruluğu oranları ise sırasıyla *hardlim-120* ve *hardlim-50* ile ve %82.93 ve %82.12 ortalama doğruluk oranlarındadır. Bu sonuçlara göre en yüksek ortalama doğruluk oranına sahip olan *hardlim-75* aktivasyon fonksiyonu ve gizli nöron sayısı kullanılarak her biri 14 satırdan oluşan iki farklı test verisi üzerinde çalışmalar yapılmıştır. Test verileri, 14.01.2019-01.01.2019 ve 14.02.2019-01.02.2019 arasındaki verilerdir. Test verilerinin her satırında yedi adet giriş verisi ile bu giriş verilerine karşılık gelen HKİ değerleri bulunmaktadır. Ayrıca, HKİ değerlerine karşılık gelen ve bu çalışmada yapılan matematiksel sınıflandırma değerleri de bulunmaktadır. Bu değerler EK bölümündeki Tablo E2'de verilmiştir.

Buna göre en yüksek ortalama doğruluk oranına sahip olan *hardlim-75* aktivasyon fonksiyonu ve nöron sayısı 2019 yılının Ocak ayının ilk 14 günü verilerini içeren test verilerine uygulandığında elde edilen tahmini HKİ sınıflandırma değerleri Tablo 2'deki gibi olmuştur. Görüleceği üzere, 14 satırdan oluşan sınıflandırmanın 12 tane doğru sınıflandırmasına denk gelen %85.71'lik bir başarım elde edilmiştir. Bu oran yukarıda elde edilen %85.37 başarım oranına yakın bir değerdir. Test süresi 0.0004888 saniye

olarak elde edilmiştir. Bu sonuçlara göre, 14 adet sınıflandırma değerinin 12 tanesinin AÖM tarafından doğru tahmin edildiği görülmektedir.

Benzer şekilde, 2019 yılının Şubat ayının ilk 14 günlük test verileri kullanıldığında ise, AÖM yine % 85.71'lik bir başarım elde etmiştir. Test süresi 0.0003357 saniye olarak elde edilmiştir. Her iki test verisi için, hem matematiksel HKİ sınıflandırma değerleri hem de AÖM tarafından yapılan tahmini HKİ sınıflandırma değerleri Tablo 2'de gösterilmektedir.

Tablo 2: AÖM Test Sonuçlarının Karşılaştırılması (Ham Veriler)

Tarih	Matematiksel Sınıflandırma	Tahmini AÖM Sınıflandırması	Tarih	Matematiksel Sınıflandırma	Tahmini AÖM Sınıflandırması
14.01.2019	2	2	14.02.2019	1	1
13.01.2019	2	2	13.02.2019	2	2
12.01.2019	2	2	12.02.2019	2	2
11.01.2019	2	2	11.02.2019	1	2
10.01.2019	2	2	10.02.2019	1	1
09.01.2019	1	1	09.02.2019	1	2
08.01.2019	2	2	08.02.2019	2	2
07.01.2019	2	2	07.02.2019	2	2
06.01.2019	1	2	06.02.2019	2	2
05.01.2019	2	2	05.02.2019	2	2
04.01.2019	2	1	04.02.2019	2	2
03.01.2019	1	1	03.02.2019	2	2
02.01.2019	2	2	02.02.2019	2	2
01.01.2019	1	1	01.02.2019	2	2

2019 yılı Ocak ayı test verilerinde iki adet yanlış HKİ sınıflandırma tahmini yapılmıştır. Matematiksel sınıflandırmaya göre 04.01.2019 tarihinde matematiksel HKİ sınıfı 2 iken, AÖM tarafından 1 olarak tahmin edilmiştir. 06.01.2019 tarihinde ise HKİ değeri matematiksel olarak 1. sınıfta iken, AÖM tarafından 2. sınıf olarak tahmin edilmiştir. Bunun nedeni, her iki tarihte hesaplanan HKİ değerlerinin sınır değer olan 50 değerine çok yakın olması ve AÖM algoritmasının bunu tahmin edememesidir. Aynı şekilde, 2019 yılı Şubat ayı test verilerinde de iki yanlış HKİ sınıflandırma tahmini yapılmıştır. Matematiksel sınıflandırma değerleri 09.02.2019 ve 11.02.2019 tarihlerinde 1 iken, AÖM tarafından 2. sınıf olarak tahmin edilmiştir.

3.3. Normalize Veriler ile AÖM Çalışması

Bu bölümde, normalize edilmiş verilerle HKİ sınıfının değerlerinin tahmin edilmesi amaçlanmıştır. Buna göre, normalize edilmiş verilerin 3 farklı aktivasyon fonksiyonuna ve 6 farklı gizli nörona bağımlılığı araştırılmıştır. Sonucun güvenilirliği için veriler kendi aralarında yine 3 kat çapraz doğrulama tekniğine tabi tutulmuşlardır. Elde edilen değerler Tablo 3'teki gibi olmuştur.

Tablo 3: Farklı Aktivasyon Fonksiyonu ve Nöron Sayısında Test Değerleri (Normalize Veri)

Aktivasyon kodu - gizli nöron sayısı	Test Süresi (sn.)	Ortalama Test Doğruluğu (%)
sig-10	0.0003916	73.98
sin-10	0.0004657	76.42
hardlim-10	0.0005113	43.09
sig-25	0.0004153	62.60
sin-25	0.0004465	85.37
hardlim-25	0.0004468	48.78
sig-50	0.0006831	61.79
sin-50	0.0006360	59.35
hardlim-50	0.0006188	59.35
sig-75	0.0006147	43.09
sin-75	0.0006132	50.41
hardlim-75	0.0005662	59.35
sig-100	0.0036135	46.34
sin-100	0.0006133	53.66
hardlim-100	0.0005662	57.72
sig-120	0.0007889	46.34
sin-120	0.0006365	47.15
hardlim-120	0.0005474	47.97

Tablo 3'ten görülebileceği gibi, en yüksek test doğruluğu oranı %85.37'lik başarım oranı ile sin-25'te elde edilmiştir. Bu doğrultuda, normalize edilmiş test verileri üzerinde yapılan çalışmada, sin-25 aktivasyon fonksiyonu ve gizli nöron sayısı kullanılmıştır. 14.01.2019-01.01.2019 ve 14.02.2019-01.02.2019 tarihleri arasındaki 7 adet normalize edilmiş giriş verisi ile bu giriş verilerine karşılık gelen HKİ değerleri ve matematiksel sınıflandırma değerleri EK bölümünde Tablo E3'te verilmiştir. 2019 yılı Ocak ayına ait normalize edilmiş 14 günlük test verileri için AÖM algoritması (sin-25) uygulandığında 0.0004551 saniyede %85.71 oranında başarım elde edilmiştir. 12 adet HKİ sınıflandırma değeri doğru tahmin edilmiştir.

Benzer şekilde, 2019 yılı Şubat ayına ait normalize edilmiş 14 adet test verisi kullanıldığında, AÖM algoritması (sin-25) %71.43'lük başarım göstermiştir. Tahmin süresi ise 0.0003926 saniye olmuştur. Sadece 10 adet HKİ verisi doğru olarak tahmin edilebilmiştir. Tablo 4, normalize edilmiş verilere karşılık gelen matematiksel HKİ sınıflandırma değerlerini ve AÖM tarafından yapılan HKİ sınıflandırma değerlerini göstermektedir.

Tablo 4: AÖM Test Sonuçlarının Karşılaştırılması (Normalize Veriler)

Tarih	Matematiksel Sınıflandırma	Tahmini AÖM Sınıflandırması	Tarih	Matematiksel Sınıflandırma	Tahmini AÖM Sınıflandırması
14.01.2019	2	2	14.02.2019	1	4
13.01.2019	2	2	13.02.2019	2	2
12.01.2019	2	2	12.02.2019	2	2
11.01.2019	2	2	11.02.2019	1	2
10.01.2019	2	2	10.02.2019	1	2
09.01.2019	1	1	09.02.2019	1	2
08.01.2019	2	2	08.02.2019	2	2
07.01.2019	2	2	07.02.2019	2	2
06.01.2019	1	2	06.02.2019	2	2
05.01.2019	2	2	05.02.2019	2	2
04.01.2019	2	1	04.02.2019	2	2
03.01.2019	1	1	03.02.2019	2	2
02.01.2019	2	2	02.02.2019	2	2
01.01.2019	1	1	01.02.2019	2	2

2019 yılı Ocak ayı normalize edilmiş test verilerinde de, ham verilerde olduğu gibi iki adet yanlış HKİ sınıflandırma tahmini yapılmıştır. Ham verilerle yapılan tahmin çalışmasında %85.71 oranında başarım elde edilirken, normalize edilmiş verilerle yapılan tahmin çalışmasında da % 85.71 oranında başarım elde edilmiştir. Bu başarımların süresi ham verilerle yapıldığında 0.0004888 saniye iken normalize edilmiş verilerle yapıldığında 0.0004551 saniye olmuştur. Bölüm 3.5'te de belirtildiği üzere verilerin normalize edilmesinin tahmin süresini kısmen de olsa azalttığı görülmektedir. 2019 yılı Şubat ayı normalize edilmiş test verileri ile yapılan tahmin çalışmasında AÖM algoritması 14 adet verinin 10 tanesini doğru tahmin ederek %71.43 oranında başarım elde etmiştir. Bu oran ham veriler ile yapılan tahmin çalışmasında %85.71 olarak elde edilmişti.

3.4. Ham Veriler ile YSA Çalışması

Bu kısımda yapılan çalışmada da aynı eğitim ve test verileri kullanılmıştır. Yine 3 farklı eğitim fonksiyonu ve 6 farklı gizli nöron kullanılmıştır. Bu değerler EK bölümündeki Tablo E4'te verilmiştir. İlk karşılaştırma aşamasında kullanılan gizli nöron sayısı yine 50 olarak seçilmiştir. 50 nöron seçilmesinin nedeni, YSA algoritmasının ilk aşamada nasıl tepki verdiğini görmek ve asıl test verilerinde kullanılması gereken gizli nöron sayısını belirlemektir. Ortalama sonuçlar elde etmek için 3 kat çapraz doğrulama tekniği uygulanmıştır. Yapılan çalışma sonucunda elde edilen testin doğruluk oranları incelendiğinde, en yüksek doğruluk oranının %95.12 ile traincgp eğitim fonksiyonunda, daha sonra ise %92.68 doğruluk oranı ile trainbfg eğitim fonksiyonunda gerçekleşmiştir. Test doğruluğu oranı %90.24 olan iki eğitim fonksiyonu bulunmaktadır. Trainbr eğitim fonksiyonunun sonuca ulaşmada çok yüksek zaman gerektirmesinden dolayı aşağıdaki test çalışmalarında kullanılmamıştır. Bu doğrultuda çalışma-lar en yüksek test doğruluğu oranına sahip olan traincgp, trainbfg ve traingd eğitim fonksiyonları ve bu fonksiyonların farklı gizli nöron sayılarında yapılmıştır. Yapılan çalışmalar sonucunda elde edilen değerler Tablo 5'teki gibidir.

Tablo 5: Farklı Eğitim Fonksiyonu ve Nöron Sayısında Test Değerleri (Ham Veri)

Aktivasyon kodu ve gizli nöron sayısı	**Test Süresi (sn.)**	**Ortalama Test Doğruluğu (%)**	**Aktivasyon kodu ve gizli nöron sayısı**	**Test Süresi (sn.)**	**Ortalama Test Doğruluğu (%)**
traincgp-10	0.1729	82.93	traincgp-75	0.2184	43.90
trainbfg-10	0.2642	85.37	trainbfg-75	2.6556	80.49
traingd-10	2.2404	95.12	traingd-75	3.5547	90.24
traincgp-25	0.3743	90.24	traincgp-100	0.6780	87.80
trainbfg-25	0.4836	90.24	trainbfg-100	6.5453	56.10
traingd-25	2.5513	97.56	traingd-100	0.1497	4.88
traincgp-50	0.1908	63.41	traincgp-120	0.2787	53.66
trainbfg-50	1.3862	82.93	trainbfg-120	18.8743	80.49
traingd-50	2.9794	92.68	traingd-120	2.0271	78.05

Tablo 5'te görülebileceği gibi, en yüksek test doğruluğu oranları traingd eğitim fonksiyonunun 10, 25 ve 50 gizli nöron sayılarında elde edilmiştir. En yüksek ortalama test doğruluğu oranları ise traingd-25 ile %97,56 oranında elde edilmiştir. Her iki test verisine ait tüm ham değerler EK bölümünde Tablo E2'de verilmiştir.

Buna göre, YSA algoritması traingd-25 aktivasyon fonksiyonu ve gizli nöron sayısında 123 satırdan oluşan eğitim verilerine ve 2019 yılı Ocak ayının ilk 14 günlük ham test verilerine uygulandığında %71.43 oranında başarım elde edilmiştir. Test süresi 2.4109 saniye olmuştur. Benzer şekilde, YSA algoritması aynı aktivasyon fonksiyonu ve nöron sayısında aynı eğitim verilerine ve 2019 yılı Şubat ayının ilk 14 günlük ham test verilerine uygulandığında ise %85.71 oranında başarım elde edilmiştir. Test süresi 2.5841 saniye olmuştur. Her iki test verisi için YSA tahmin sonuçları Tablo 6'daki gibidir.

Tablo 6: YSA Test Sonuçlarının Karşılaştırılması (Ham Veriler)

Tarih	Matematiksel Sınıflandırma	Tahmini YSA Sınıflandırması	Tarih	Matematiksel Sınıflandırma	Tahmini YSA Sınıflandırması
14.01.2019	2	1	14.02.2019	1	1
13.01.2019	2	2	13.02.2019	2	2
12.01.2019	2	2	12.02.2019	2	2
11.01.2019	2	2	11.02.2019	1	1
10.01.2019	2	2	10.02.2019	1	1
09.01.2019	1	2	09.02.2019	1	2
08.01.2019	2	2	08.02.2019	2	2
07.01.2019	2	2	07.02.2019	2	2
06.01.2019	1	1	06.02.2019	2	2
05.01.2019	2	2	05.02.2019	2	2
04.01.2019	2	1	04.02.2019	2	2
03.01.2019	1	2	03.02.2019	2	2
02.01.2019	2	2	02.02.2019	2	2
01.01.2019	1	1	01.02.2019	2	1

2019 yılı Ocak ayına ait 14 günlük ham test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında 0.0004888 saniyelik sürede %85.71'lik başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında 2.4109 saniyelik sürede %71.43'lik başarım elde edilmiştir. Aynı yılın Şubat ayına ait 14 günlük ham test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında ise 0.0003357 saniyelik sürede %85.71'lik başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında 2.5851 saniyelik sürede yine %85.71'lik başarım elde edilmiştir. Bu sonuçlara göre ham veriler ile yapılan sınıflandırma çalışmalarında AÖM algoritmasının daha başarılı olduğu sonucuna ulaşılabilmektedir.

3.5. Normalize Veriler ile YSA Çalışması

Bu bölümde, yine aynı verilerin normalize edilmesi durumunda YSA algoritmasının HKİ sınıfını tahmin etmesi üzerine bir çalışma yapılmıştır. Buna göre, normalize edilmiş verilerin 3 aktivasyon fonksiyonunun 6 farklı gizli nöron sayısına bağımlılığı araştırılmış ve Tablo 7 değerleri elde edilmiştir.

Tablo 7: Farklı Eğitim Fonksiyonu ve Nöron Sayısında Test Değerleri (Normalize Veriler)

Aktivasyon kodu ve gizli nöron sayısı	Test Süresi (sn.)	Ortalama Test Doğruluğu (%)	Aktivasyon kodu ve gizli nöron sayısı	Test Süresi (sn.)	Ortalama Test Doğruluğu (%)
traincgp-10	0.1734	63.41	traincgp-75	0.2647	68.29
trainbfg-10	0.2611	68.29	trainbfg-75	2.1783	70.73
traingd-10	2.4944	68.29	traingd-75	0.1590	0.00
traincgp-25	0.1443	26.83	traincgp-100	0.3978	68.29
trainbfg-25	0.4678	68.29	trainbfg-100	22.4158	75.61
traingd-25	1.7159	82.93	traingd-100	0.1317	0.00
traincgp-50	0.2588	78.05	traincgp-120	0.3530	68.69
trainbfg-50	1.8703	65.85	trainbfg-120	21.9830	63.41
traingd-50	3.5596	80.49	traingd-120	0.1367	0.00

Tablo 7'den görülebileceği gibi, en yüksek test doğruluğu oranı traingd-25 ile % 82.93 performans oranında elde edilmiştir. Ardından gelen en yüksek performans oranları ise traingd-50'de % 80.40 ve traincgp-50'de% 78.05 olmuştur. Traingd-25, 123 satırdan oluşan normalize edilmiş eğitim verilerine

ve 2019 yılı Ocak ayına ait normalize edilmiş test verilerine uygulandığında, 1.0494 saniye süre ile % 64.29 başarım elde edilmiştir. Aynı şekilde, traingd-25 eğitim fonksiyonu ve nöron sayısı, normalize edilmiş eğitim verilerine ve 2019 yılı Şubat ayına ait 14 günlük normalize edilmiş test verilerine uygulandığında, 0.4575 saniyede % 42.86 başarı elde edilmiştir. Her iki test verisi için elde edilen tahmin değerleri Tablo 8'de verilmiştir.

Tablo 8: YSA Test Sonuçlarının Karşılaştırılması (Normalize Veriler)

Tarih	Matematiksel Sınıflandırma	Tahmini YSA Sınıflandırması	Tarih	Matematiksel Sınıflandırma	Tahmini YSA Sınıflandırması
14.01.2019	2	2	14.02.2019	1	1
13.01.2019	2	2	13.02.2019	2	1
12.01.2019	2	2	12.02.2019	2	1
11.01.2019	2	2	11.02.2019	1	1
10.01.2019	2	2	10.02.2019	1	1
09.01.2019	1	2	09.02.2019	1	1
08.01.2019	2	1	08.02.2019	2	1
07.01.2019	2	2	07.02.2019	2	1
06.01.2019	1	0	06.02.2019	2	1
05.01.2019	2	2	05.02.2019	2	1
04.01.2019	2	1	04.02.2019	2	2
03.01.2019	1	2	03.02.2019	2	2
02.01.2019	2	2	02.02.2019	2	1
01.01.2019	1	1	01.02.2019	2	1

Bu sonuçlara göre AÖM algoritmasında olduğu gibi YSA algoritmasında da ham veriler ile yapılan çalışmaya ait başarım oranlarının normaliz edilmiş veriler ile yapılan başarım oranından daha yüksek olduğu görülmüştür. Bu bağlamda sadece ham veriler ile çalışma yapılmasının sonuçların tahminini yapmada yeterli olacağı sonucuna ulaşılmıştır.

2019 yılı Ocak ayına ait 14 günlük normalize edilmiş test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında 0.0004551 saniyelik sürede % 85.71'lik başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında 1.0494 saniyelik sürede % 64.29'luk başarım elde edilmiştir. Aynı yılın Şubat ayına ait 14 günlük normalize edilmiş test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında ise 0.0003926 saniyelik sürede % 71.43'lük başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında 0.4575 saniyelik sürede yine % 42.86'lık başarım elde edilmiştir. Bu sonuçlara göre ham veriler olduğu gibi normalize edilmiş veriler ile yapılan sınıflandırma çalışmalarında da AÖM algoritmasının daha başarılı olduğu sonucuna ulaşılabilmektedir.

Yukarıda literatür taraması yapılan kısımda belirtilen çalışmalar incelendiğinde genellikle PM_{10}, SO_2, CO parametreleri kullanılmışken bu çalışmada ek olarak sıcaklık, nem, basınç ve rüzgâr hızı parametreleri de HKİ değerinin belirlenmesinde giriş parametresi olarak dikkate alınmıştır. Meteorolojik parametrelerin HKİ değerini nasıl etkileyebileceğine dair analizler yapılmıştır. Analiz sonucuna göre HKİ ile en güçlü ilişkilerin atmosferik parametrelerden PM_{10} ile, meteorolojik parametrelerden ise basınç ile olduğu sonucuna ulaşılmıştır. Ayrıca ham verilerin yanında normalize veriler ile de çalışmalar yapılmıştır. Farklı modeller kullanılan çalışmalarda Biancofiore vd. (2017), %95, Mekparyup ve Saithanu (2020), %90 ve Dragomir (2010) %65.52 oranlarında başarımlar elde etmişlerdir. Zhang ve Ding (2017) tarafından yapılan çalışmada da atmosferik ve meteorolojik parametreler dikkate alınmıştır. AÖM ile FFANN-BP algoritmalarını karşılaştırılmıştır. AÖM'nin FFANN-BP algoritmasından daha başarılı sonucuna ulaşmışlardır. Bu çalışmada ise AÖM ile yapılan çalışmalarda ham veriler için %85.71, normalize veriler için %71.43 başarım elde edilmiştir.

4. Sonuç

Bu çalışmada HKİ sınıf değerlerinin AÖM ve YSA algoritmaları ile tahmin edilmesi amaçlanmıştır. Elde edilen sonuçlara göre 2019 yılı Ocak ayına ait 14 günlük ham test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında %85.71'lik başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında %71.43'lik başarım elde edilmiştir. Aynı yılın Şubat ayına ait 14 günlük ham test verilerine ait HKİ sınıflarının hem AÖM hem de algoritması ile yapılan tahmin çalışmalarında ise %85.71'lik başarımlar elde edilmiştir. Bu sonuçlara göre ham veriler ile yapılan sınıflandırma çalışmalarında AÖM algoritması ile daha başarılı sonuçlar elde edilmiştir. 2019 yılı Ocak ayına ait 14 günlük normalize edilmiş test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında ise %85.71'lik başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında %64.29'luk başarım elde edilmiştir. Şubat ayına ait 14 günlük normalize edilmiş test verilerine ait HKİ sınıflarının AÖM algoritması ile yapılan tahmin çalışmasında ise %71.43'lük başarım oranı elde edilmişken, YSA algoritması ile yapılan tahmin çalışmasında %42.86'lık başarım elde edilmiştir. Sonuç olarak, bu çalışmada AÖM ile yapılan çalışmalarda ham verilerde %85.71, normalize verilerde %71.43 başarım gösterilmiştir. Hem meteorolojik hem de atmosferik parametrelerin dikkate alındığı AÖM ve YSA algoritmaları ile yapılan bu sınıflandırma çalışmasının insan sağlığını olumsuz etkileyebilecek HKİ değerinin tahmin edilmesinde yol gösterici olacağı düşünülmektedir.

Çıkar Çatışmaları

Yazar herhangi bir çıkar çatışması beyan etmemektedir.

Referanslar

[1] Shaban, K.B., Kadri, A. and Rezk, E., 2016. Urban Air Pollution Monitoring System With Forecasting Models. IEEE Sensors Journal, 16 (8). DOI: 10.1109/JSEN.2016.2514378.

[2] Yang, Z. and Wang, J., 2017. A new air quality monitoring and early warning system: Air quality assessment and air pollutant concentration prediction. Environmental Research, https://www.sciencedirect.com/science/journal/00139351/158/supp/C158, 105-117. https://doi.org/10.1016/j.envres.2017.06.002.

[3] Karamchandani, S. and Gupta, D., 2016. Pervasive monitoring of carbon monoxide and methane using air quality prediction. 3rd International Conference on Computing for Sustainable Global Development (INDIACom).

[4] Karacı, A., 2018. Akıllı Şehir Hava Takip Sistemi ve Astım Hastaları için PM2.5 Konsantrasyonu Ölçüm Aracının Geliştirilmesi, Mühendislik Bilimleri ve Tasarım Dergisi, 6:3, 418 – 425. https://doi.org/10.21923/jesd.412665.

[5] Turalioglu, F.S., 2005. An Assessment on Variation of Sulphur Dioxide and Particulate Matter in Erzurum (Turkey). Environmental Monitoring Assessment, 104, 119–130.

[6] EPA, 2019. Birleşik Devletler Çevre Koruma Ajansı. Karbon monoksit Hava Kirliliği. https://www.epa.gov/.

[7] Rao, K.S. Devi, G.L. and Ramesh, N., 2019. Air Quality Prediction in Visakhapatnam with LSTM based Recurrent Neural Networks. International Journal of Intelligent Systems and Applications, 2, 18-24. DOI: 10.5815/ijisa.2019.02.03.

[8] Temiz, İ. and Turgut, D., 2015. Time Series Analysis and Forecasting For Air Pollution In Ankara: A Box-Jenkıns Approach. Alphanumeric Journal, 3 (2), 131-138. https://doi.org/10.17093/aj.2015.3.2.5000148347.

[9] Sevinç, E., 2022. An empowered AdaBoost algorithm implementation: A COVID-19 dataset study. Computers & Industrial Engineering, 165:107912. doi: 10.1016/j.cie.2021.107912.

[10] Cihan, P., Ozel, H., & Ozcan, H. K. (2021). Modeling of atmospheric particulate matters via artificial intelligence methods. Environmental Monitoring and Assessment, 193 (5), 1-15. https://doi.org/10.1007/s10661-021-09091-1.

[11] Baran, B., 2021. Air Quality Index Prediction in Besiktas District by Artificial Neural Networks and K Nearest Neighbours. Journal of Engineering Sciences and Design, 9(1): 52 – 63. DOI: 10.21923/jesd.671836.

[12] Shishegaran, A., Saeedi, M., Kumar, A. and Ghiasinejad, H., 2020. Prediction of air quality in Tehran by developing the nonlinear ensemble model. Journal of Cleaner Production, 259, 120825. https://doi.org/10.1016/j.jclepro.2020.120825.

[13] Wang, J., Du, P., Hao, Y., Ma, X., Niu, T. and Yang, W., 2020. An innovative hybrid model based on outlier detection and correction algorithm and heuristic intelligent optimization algorithm for daily air quality index forecasting. Journal of Environmental Management, 255, 109855. https://doi.org/10.1016/j.jenvman.2019.109855.

[14] Baran, B., 2019. Aşırı Öğrenme Makineleri ile Rüzgar Hızına Bağlı Enerji Tahmini: Malatya Örneği. 1. Ulusal

Mühendislik ve Teknoloji Kongresi (UMTK). 56-62.

[15] Liu, H., Li, Q., Yu, D. and Gu, Y., 2019. Air Quality Index and Air Pollutant Concentration Prediction Based on Machine Learning Algorithms. Applied Sciences, 9 (4069), 1-9. https://doi.org/10.3390/app9194069.

[16] Sevinç, E., 2019. A Novel Evolutionary Algorithm for Data Classification Problem With Extreme Learning Machines, IEEE Access, 7:122419-122427, doi: 10.1109/ACCESS.2019.2938271.

[17] Zou, Z., Cai, T. and Cao, K., 2019. An urban big data-based air quality index prediction: A case study of routes planning for outdoor activities in Beijing. Environment and Planning B: Urban Analytics and City Science. https://doi.org/10.1177/2399808319862292.

[18] Bai, Y., Li, Y., Wang, X., Xie, J. and Li, C., 2016. Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions. Atmospheric Pollution Research, 7 (3), 557-566. https://doi.org/10.1016/j.apr.2016.01.004.

[19] Baran, B., 2019. Prediction of Air Quality Index by Extreme Learning Machines. 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 1-8. doi: 10.1109/IDAP.2019.8875910.

[20] Zhang, J. and Ding, W., 2017. Prediction of Air Pollutants Concentration Based on an Extreme Learning Machine: The Case of Hong Kong. International Journal of Environmental Research and Public Health, 14 (2), 114. doi: 10.3390/ijerph14020114.

[21] Sevinç, E., 2018. Activation functions in single hidden layer feed-forward neural networks. Selcuk University Journal of Engineering Sciences, 1-13.

[22] Zhu, S., Lian, X., Liu, H., Hu, J., Wang, Y.Y. and Che, Z., 2017. Daily air quality index forecasting with hybrid models: A case in China. Environmental Pollution, 231 (2), 1232-1244. DOI: 10.1016/j.envpol.2017.08.069

[23] Peng, H., Lima, A.R., Teakles, A., Jin, J., Cannon, A.J. and Hsieh, W.W., 2017. Evaluating hourly air quality forecasting in Canada with nonlinear updatable machine learning methods. Air Quality, Atmosphere, & Health, 10 (2), 195-211.

[24] Jose, L.A., 2017. Probabilistic forecasting for extreme NO2 pollution episodes. Environmental Pollution, 229, 321-328. https://doi.org/10.1016/j.envpol.2017.05.079.

[25] Patricio, P. and Ernesto, G., 2016. Forecasting hourly PM2.5 in Santiago de Chile with emphasis on night episodes. Atmospheric Environment, 124, 22-27. https://doi.org/10.1016/j.atmosenv.2015.11.016.

[26] Avşar, E., 2015. Balıkesir İli Burhaniye İlçesi (İskele Mahallesi) hava kalitesinin Değerlendirilmesi. Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 4(1): 68-82. https://doi.org/10.17798/beufen.40291

[27] Vong, CM., Fai Ip, W., Wong, PK. and Chiu, CC., 2014. Predicting minority class for suspended particulate matters level by extreme learning machine. Neurocomputing, 128, 136-144. https://doi.org/10.1016/j.neucom.2012.11.056.

[28] Moustris, K.P., Ziomas, I.C. and Paliatsos, A.G., 2010. 3-Day-Ahead Forecasting of Regional Pollution Index for the Pollutants NO2, CO, SO2, and O3 Using Artificial Neural Networks in Athens, Greece. Water, Air, & Soil Pollution, 209 (1–4), 29–43.

[29] Biancofiore, F., Busilacchio, M., Verdecchia, M., Tomassetti, B., Aruffo, E., Bianco, S., Di Tommaso, S., Colangeli, C., Rosatelli, G. and Di Carlo, P., 2017. Recursive neural network model for analysis and forecast of PM10 and PM2.5. Atmospheric Pollution Research, 8, 652–659. https://doi.org/10.1016/j.apr.2016.12.014.

[30] Mekparyup, J. and Saithanu, K., 2020. Air Quality Index Prediction in the Eastern Regions of Thailand with Accuracy of Neural Networks. International Journal of Applied Engineering Research, 15 (5), 436-444.

[31] Liu, B.C., Binaykia, A., Chang, P.C., Tiwari, M.K. and Tsao, C.C., 2017. Urban air quality forecasting based on multi-dimensional collaborative Support Vector Regression (SVR): A case study of Beijing-Tianjin-Shijiazhuang. PLoSONE, 12 (7), e0179763. https://doi.org/10.1371/journal.pone.0179763.

[32] Ganesh, S.S., Arulmozhivarman, P. and Tatavarti, R., 2017. Forecasting Air Quality Index Using an Ensemble of Artificial Neural Networks and Regression Models. Journal of Intelligent Systems. https://doi.org/10.1515/jisys-2017-0277.

[33] Saatcioglu, T., Alp, K., Hanedar, A. and Avşar, E., (2011). Effect of the marmaray project on air pollution in Istanbul: An ive model application, Fresenius Environmental Bulletin. 20 (9): 2340-2349,

[34] Dragomir, E.G., 2010. Air Quality Index Prediction using K-Nearest Neighbor Technique. Bulletin of PG University of Ploiesti, Series Mathematics, Informatics, Physics, 62 (1), 103-108.

[35] Jiao, Y., Wang, Z. and Zhang, Y., 2019. Prediction of Air Quality Index Based on LSTM. IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing-China, 17-20. doi: 10.1109/ITAIC.2019.8785602.

[36] Kadilar, G.O and Kadilar, C., 2017. Assessing air quality in Aksaray with time series analysis. AIP Conference Proceedings, 1833, 020112.

[37] Avşar, E., Hanedar, A., Toroz, I., Alp, K. and Kaynak, B., (2010). Investigation of PM_{10} Concentrations And Noise Levels of The Road Sweepers Operating In Istanbul-Turkey: A Case Study, Fresenius Environmental Bulletin 19 (9b).

[38] AQI Calculator, 2018. Hava Kalitesi İndeksi Hesaplama Aracı. https://app.cpcbccr.com/ccr_docs/AQI%20-Calculator.xls.

[39] YSAb, 2019. Yapay Sinir Ağları. https://blogs.mathworks.com/loren/2015/08/04/artificial-neural-networks-for-beginners/.

[40] Time and Data, 2019. Ankara İli 2018-2019 Hava Durumu Bilgisi.https://www.timeanddate.com/weather/turkey/ankara/historic.

[41] Taşdelen, B., 2019. Korelasyon ve Regresyon Analizi. Erişim adresi: https://docplayer.biz.tr/47627312-Korelasyon-ve-regresyon-analizi-doc-dr-bahar-tasdelen.html.

[42] Huang, G.B., Zhu, Q.Y. and Siew, C.K., 2006. Extreme Learning Machine: Theory and Applications. Neurocomputing, 70, 489-501. https://doi.org/10.1016/j.neucom.2005.12.126.

[43] Kaya, Y., 2014. A Fast Intelligent Diagnosis System For Thyroid Diseases Based On Extreme Learning Machine. Bilim ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik, 15 (1). https://doi.org/10.18038/btd-a.89202.

[44] Baran, B., 2019. Sınır Değerler Arasında Kalan Evsel Atıksu Numune Analizi Sonucunun Aşırı Öğrenme Makineleri ile Sınıflandırılması. Mühendislik Bilimleri ve Tasarım Dergisi, 7 (1), 18 – 25. https://doi.org/10.21923/jesd.457085.

[45] YSAa, 2019. Yapay Sinir Ağları. http://kod5.org/yapay-sinir-aglari-ysa-nedir/.

[46] İnalpulat, M., Kızıl, Ü., Bilgücü, E. and Genç, L., 2016. E-Nose Identification of Milk Somatic Cell Count, Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 2:1, 22-35.

[47] Saplıoğlu, K., Küçükerdem Öztürk, T.S. and Şenel, FA., 2020. Eksik Hidrolojik Verilerin Simbiyotik Organizmalar Arama Algoritması ile Tahmini, Çanakkale Onsekiz Mart Üniversitesi Fen Bilimleri Enstitüsü Dergisi Açık Erişim, 6:1, 93-104. https://doi.org/10.28979/comufbed.628846.

[48] FFBPNNa, 2019. Feed-forward Back Propagation Neural Network.https://stackoverflow.com/questions/28403782/what-is-the-difference-between-back-propagation-and-feed-forward-neural-network.

[49] Yavuz, S. and Deveci M., 2012. İstatiksel Normalizasyon Tekniklerinin Yapay Sinir Ağın Performansına Etkisi. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 40 (2), 167-187.

EK

Bu çalışmaya ait ek veriler aşağıdaki gibidir.

Tablo E1: Farklı Aktivasyon Fonksiyonlarında Test Süresi ve Test Doğruluk Değerleri (50 nöron)

Aktivasyon Fonksiyonu	Test Süresi (sn.)	Test Doğruluğu (%)
sig	0.001073	71.54
sin	0.000582	30.90
hardlim	0.000520	76.42

Tablo E2: Parametreler ve Matematiksel Sınıflandırma Değerleri (ham veri / 14.01.2019-01.01.2019 ve 14.02.2019-01.01.2019)

14.01.2019 - 01.01.2019									**14.02.2019 - 01.02.2019**								
PM_{10}	**SO_2**	**CO**	**Sıcaklık**	**Nem (%)**	**Basınç (mbar)**	**Rüzgar hızı (kmh)**	**Hesaplanan HKİ**	**Matematiksel Sınıf**	**PM_{10}**	**SO_2**	**CO**	**Sıcaklık**	**Nem (%)**	**Basınç (mbar)**	**Rüzgar hızı (kmh)**	**Hesaplanan HKİ**	**Matematiksel Sınıf**
26.18	2.94	1360	6	77	1002	11	68	2	37.36	11.78	574	7	68	1014	20	37	1
48.53	5.16	1502	4	86	1005	3	75	2	56.06	12.38	1305	11	54	1017	10	65	2
35.06	5.85	1400	7	66	1010	6	70	2	53.58	11.33	1378	13	46	1015	15	69	2
47.35	10.05	1560	7	64	1018	9	78	2	31.67	9.05	978	9	57	1018	13	49	1
71.22	16.49	1956	1	60	1019	11	98	2	28.44	10.41	922	8	64	1018	9	46	1
18.16	10.19	977	-3	63	1017	9	49	1	30.85	10.9	1042	8	76	1015	6	50	1
40.45	8.47	1063	-1	62	1008	13	53	2	32.64	9.94	1116	8	85	1013	7	55	2
35.61	14.14	1280	-1	74	1013	4	64	2	40.16	12.37	1258	9	81	1012	12	63	2
32.57	8.86	983	-1	92	999	12	49	1	59.6	18.08	1353	11	68	1017	8	68	2
47.15	6.67	1389	2	87	1009	9	70	2	69.05	15	1340	16	37	1021	10	69	2
27.32	6.47	1014	5	55	1017	23	51	2	79.38	13.81	1816	16	37	1023	6	90	2
18.8	5.7	956	8	68	1010	11	48	1	66.74	10.87	1543	13	42	1022	4	77	2
49.42	10.94	1441	5	80	1009	3	72	2	50.84	14.1	1545	9	55	1021	6	78	2
31.33	9.38	936	-1	67	1019	18	47	1	16.93	10.49	1077	7	64	1016	31	54	2

Tablo E3: Parametreler ve Matematiksel Sınıflandırma Değerleri (normalize veri / 14.01.2019-01.01.2019 ve 14.02.2019-01.01.2019)

14.01.2019 - 01.01.2019									14.02.2019 - 01.02.2019								
PM_{10}	SO_2	CO	Sıcaklık	Nem (%)	Basınç (mbar)	Rüzgar hızı (kmh)	Hesaplanan HKİ	Matematiksel Sınıf	PM_{10}	SO_2	CO	Sıcaklık	Nem (%)	Basınç (mbar)	Rüzgar hızı (kmh)	Hesaplanan HKİ	Matematiksel Sınıf
0,07	0,011	0,442	0,237	0,810	0,103	0,003	68	2	0,125	0,097	0,035	0,263	0,696	0,517	0,645	37	1
0,18	0,032	0,516	0,184	0,924	0,207	0,007	75	2	0,220	0,103	0,413	0,368	0,519	0,621	0,323	65	2
0,11	0,039	0,463	0,263	0,671	0,379	0,012	70	2	0,208	0,093	0,452	0,421	0,418	0,552	0,484	69	2
0,18	0,080	0,546	0,263	0,646	0,655	0,021	78	2	0,096	0,071	0,244	0,316	0,557	0,655	0,419	49	1
0,30	0,143	0,751	0,105	0,595	0,690	0,022	98	2	0,079	0,084	0,215	0,289	0,646	0,655	0,290	46	1
0,03	0,082	0,244	0,000	0,633	0,621	0,020	49	1	0,091	0,089	0,277	0,289	0,797	0,552	0,194	50	1
0,14	0,065	0,288	0,053	0,620	0,310	0,010	53	2	0,101	0,079	0,316	0,289	0,911	0,483	0,226	55	2
0,12	0,120	0,401	0,053	0,772	0,483	0,016	64	2	0,139	0,103	0,389	0,316	0,861	0,448	0,387	63	2
0,10	0,069	0,247	0,053	1,000	0,000	0,000	49	1	0,239	0,159	0,439	0,368	0,696	0,621	0,258	68	2
0,17	0,047	0,457	0,132	0,937	0,345	0,011	70	2	0,287	0,129	0,432	0,500	0,304	0,759	0,323	69	2
0,07	0,045	0,263	0,211	0,532	0,621	0,020	51	2	0,340	0,117	0,678	0,500	0,304	0,828	0,194	90	2
0,03	0,038	0,233	0,289	0,696	0,379	0,012	48	1	0,275	0,088	0,537	0,421	0,367	0,793	0,129	77	2
0,19	0,089	0,484	0,211	0,848	0,345	0,011	72	2	0,194	0,120	0,538	0,316	0,532	0,759	0,194	78	2
0,09	0,074	0,223	0,053	0,684	0,690	0,022	47	1	0,020	0,085	0,296	0,263	0,646	0,586	1,000	54	2

Tablo E4. Farklı aktivasyon fonksiyonlarında test süreleri ve test doğruluk değerleri (50 nöron-ham veri)

Aktivasyon Fonksiyonu	Test Süresi (sn.)	Test Doğruluğu (%)	Aktivasyon Fonksiyonu	Test Süresi (sn.)	Test Doğruluğu (%)
trainlm	0.3311	63.41	traincgf	0.2612	80.49
trainbr	24.8269	90.24	traincgp	0.5598	95.12
trainbfg	5.7137	92.68	trainoss	0.2690	75.61
trainrp	0.1815	65.85	traingdx	0.7939	87.80
trainscg	0.2217	87.80	traingdm	0.2008	39.02
traincgb	0.1998	78.05	traingd	3.6422	90.24

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 2, No. 1, July 2022
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi

Learning Capabilities of AI Methodologies on Multi-Class Datasets

Ender SEVINÇ[1*]
[1]Ankara Science University, Faculty of Engineering and Architecture, Computer Engineering Department, Ankara, Türkiye;
ORCID: 0000-0001-7670-722X
* Corresponding Author: ender.sevinc@ankarabilim.edu.tr

Received: 13 April 2022; Accepted: 16 June 2022

Reference/Atıf: E. Sevinç, "Learning Capabilities of AI Methodologies on Multi-Class Datasets" Researcher, vol. 02, no. 01, pp. 19-28, Jul. 2022

Abstract

Machine Learning (ML) methods have numerous kinds of application areas up to now. Since they generally have remarkable success in learning, study areas and research field have diversified drastically. Neural networks seem to be appropriate for such a learning capability. The study discusses and examines several ML methodologies to decide the output. Since binary classification is another interesting area, the study focuses on multi-class classification problems. Datasets are chosen from a commonly known and accepted repository to avoid fakeness. Totally four different classifiers have been used to understand and know the different output classes in four different datasets. The classifiers use various arguments to work with and these will be shown and explained in detail. Two of the datasets are newly added and medium-sized, this is preferred to show that there is almost no time of execution difference among all. The system developed gives remarkable success rates and eliminates the differences among the classes using a neural networks system. It is believed that ML methods will have a wide range of application fields as researchers widen their point of view for academic studies.

Keywords: machine learning, multi-class classification, classifiers

1. Introduction

Machine learning means a neural network implementation that does mapping the input-output relationship from the known input-output pairs. The aim is to deduce the system response for unknown conditions (unknown input data). This constitutes one of the humankind's obsessions, i.e., to know the unknown. This emotion makes the human work on such algorithms or methods, machine learning is extremely promising in that sense. Supervised learning is the first step to be taken, easy to implement, and overcome the problem. The study focuses on such supervised methods.

In fact, the goal of machine learning is to choose an input-output mapping model. For example, when a model with too much capacity is overtrained, it means is ss overfitted, while a model with very less capacity is undertrained, then it is under fitted. These are the difficulties of the models used in ML algorithms.

However, ML has numerous and big advantages in making all manual processes automated. Nowadays, the world is in a big race for such automation processes. Such a power not only gives incredible capability in terms of knowledge but also provides incredible speeds and functionalities to manage heavy works previously.

Though there have been made several categorizations of ML fields, the study focuses on the most commonly used Machine Learning (ML) Algorithms. These algorithms can be listed as Linear regression, Logistic regression, Decision trees, Support Vector Machines (SVM) algorithm, Naive Bayes (NB) algorithm, KNN algorithm, K-means, Random Forest algorithm, Dimensionality reduction algorithms, Gradient boosting algorithm, and AdaBoosting algorithms, etc.

Here in the study, four among these algorithms are examined, and presented the results to show the success of ML algorithms using known datasets. These algorithms are; Decision tree, SVM, k-Nearest Neighbors algorithm (k-NN), and Naive Bayes. Decision Trees (DT) are one of the most common, advanced easy-to-use tools in decision-making. SVM, on the other hand, reaches promising results when

multi-class categorization researches. K-Nearest Neighbor is one of the oldest known approaches so as the Naïve Bayes algorithm.

There are also prominent works using the Random Forest algorithm, Gradient boosting algorithm, and AdaBoosting algorithms in the literature. The AdaBoost algorithm is a boosting technique, and one example is presented in [1]. As the algorithm runs, all the weights are re-assigned to each instance, higher weights are for incorrectly classified instances while low weights are for correct ones. The methodology works on the principle of learners growing sequentially. As a result, each subsequent learner is grown from previously grown learners. Briefly, weak learners become strong ones as the error rate increases and this goes on up to the final decision is made.

Section 2 discusses the related studies issued up to now, then in Section 3, used algorithms and classifiers are explained in detail. Their capabilities and arguments are discussed Section 4 shows the success and accuracy rates obtained by the classifiers. Finally, conclusions and future works are discussed in the last section.

2. Related Work

It is very probable to find numerous studies in the literature, artificial intelligence and even focusing on deep learning nowadays, are extremely hot and attractive. You can easily find many researchers and many studies have been issued in the field.

ML methods have a wide range of improvement and application fields. Initially, genetic algorithms (GA) have been used for taking some sort of action in a reasonable time. This attitude has been lastly explained in [2]. A good implementation of GA is presented in [3] and a data exchange through the web in [4]. Then as the needs are changes smart methodologies come to the scene, and methodologies such as Extreme Learning Machines (ELM) are used. ELM is a learning methodology for the SLFNs and its speed is remarkable. This is shown in [5] and the purposed algorithm combines GA with ELM for feature selection to reach better success rates. This study focuses on a wrapper feature selection algorithm that predicts and forms a network to map the input nodes to their output counterparts. The proposed algorithm works uniquely to reach the best solutions.

A similar approach can be seen in studies [6] and [7]. These studies are mainly about graph coloring and solving the maximum vertex weight clique problem in huge graphs. The Graph Coloring Problem (GCP) can be defined as separating and grouping the vertices of a graph into various sets to minimize the colors used. In [6], the Teaching-Learning-Based Optimization (TLBO) metaheuristic is used which is a different version of GA integrated with tabu search algorithm. Study in [7] is mainly a local tabu search algorithm that is implemented parallelly by using MPI capabilities.

However, ensemble methods have an increasing effect on ML among learning methodologies. Such methods construct a set of classifiers and then classify new data points due to their predictions. The point is to combine the predictions of several base estimators of a given learning algorithm to reach a more general and robust model for prediction.

A recent and final study is presented in [1]. Here, an Adaptive Boost Algorithm using a Decision Tree estimator is proposed. The algorithm is run on a Covid-19 dataset and a tuning process is done to a classifier, and a binary classification problem is solved with higher success rates against state-of-the-art algorithms. There are similar studies feature selection methods using ML methods. One prominent one is presented in [8], i.e., a novel Hyper Learning Binary Dragonfly Algorithm which proposes a method for making feature selection in classification algorithms.

3. Proposed Work

Four commonly known machine learning algorithms have been used in the study. These are explained below respectively.

3.1 Decision tree

A decision tree classifier is a commonly known, interesting, and used approach in multiclass classification problems. It presents a set of questions and choices. The decision tree classification algorithm is very similar to that of a binary tree. There are a root and internal nodes each of which a decision is posed, and that node can be further split into separate records which have different ongoing characteristics. Finally, the leaves refer to the classes in which the dataset is split. The Decision Tree Algorithm is a building decision tree called ID3 in [9] which employs a top-down, greedy search through the space of possible branches with no backtracking.

A decision tree has a top-down structure that starts from a root node and involves partitioning the data into subsets. Intermediate nodes simply constitute the ongoing decision processes as the process proceeds. The homogeneity of a numerical sample is calculated to evaluate and use standard deviation. For example, in the case of completely homogeneous data, the deviation is zero.

In a decision tree, all the calculations are done due to the features and ongoing processes. Figure 1 presents a graphical representation for dealing with the possible solutions to a problem. The process starts with simply asking a question that is based on the answer (Yes/No), then it further split the tree into subtrees. Figure 1 visualizes the general structure of a decision tree.

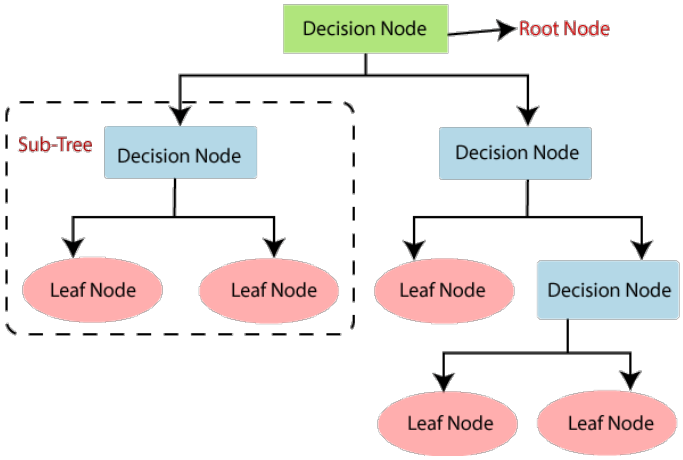


Figure 1: Sample Decision Tree

3.2 Support Vector Machines (SVM)

Support Vector Machines (SVMs) are one of the most used and promising methods in ML. It consists mainly of a set statistical method for supervised learning, and it can be applied to both classification and regression problems. The goal of SVM is to find a hyperplane among the input space of features, which has k-dimensions (k-the number of features) that distinctly classifies the data points.

One reason for the success of the SVM algorithm is that it chooses the best line to classify the data points. It decides the line that separates the data, and this line must be the furthest away from the closest data points as possible. If you minimize the bounds, the expected probability of error will be low, i.e., good generalization, and less error prone as stated in [10].

For the two-class classification problem as commonly known as separation is presented in Figure 2.

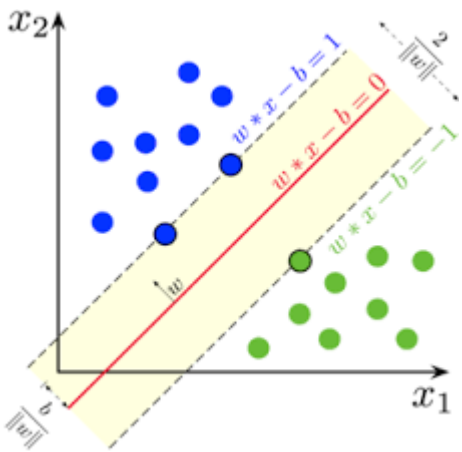


Figure 2: Support Vector Machine

3.3 k-Nearest Neighbor Algorithm

The k-Nearest Neighbors (KNN) algorithm is another commonly known and used supervised machine learning. The logic behind the KNN algorithm is simply that similar things exist in close proximity. In other words, similar things are near to each other. KNN depends on similarity which is evaluated by distance, proximity, or closeness. This calculation is done simply by evaluating the distance between points on a graph. On the other hand, the direct-line distance is a popular and known option which is also called as the Euclidean distance. This algorithm is used to solve both classification and regression problems in the literature

When it comes to classifying the data, KNN mostly and mainly uses the concept of "Majority Voting". This means that within the given range of K values, the class is chosen with the most votes. This can be barely seen in Figure 3.

However, the impact of selecting a smaller or larger K value on the model has the following

- Larger K value: The case of underfitting occurs when the value of k is increased.
- Smaller k value: The condition of overfitting occurs when the value of k is smaller.

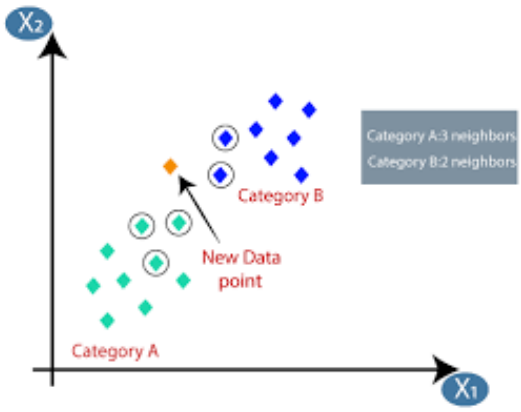


Figure 3: K-Nearest Neighbor Graph

3.4 Naive Bayes (NB)

Naïve Bayes uses Bayes Theorem as it can be understood from the name. It fundamentally uses a probabilistic approach, and this enables it to be used in a wide range of ML fields.

The Naïve Bayes algorithm is a good example among all ML algorithms for being simple. Naive Bayes' Theorem first starts with the assumption of independence predictors. This means that a particular feature in a class is irrelevant and unrelated to any other feature. Besides that, it is known to be easy to build and particularly useful for huge data sets. Along with simplicity, Naive Bayes is known to perform better than most state-of-the-art and sophisticated classification methods. as stated in [11]

The theorem works with posterior probabilities like P(c|x) from P(c), P(x), and P(x|c). Likewise, conditional probability is evaluated according to which is the probability of some sort of events. A general equation for NB is shown in Formula 1 below;

$$\mathrm{P(A|B)} = \frac{P(B|A).P(A)}{P(B)} \tag{1}$$

However, the study is a multi-class classification problem that implements the Gaussian Naive Bayes algorithm for classification. The similarity of the features is naive and calculations will be done accordingly.

4. Experiments and Implementations

The experiments are carried out on a normal laptop Windows-10 machine. It has an i7 CPU (i7-5700HQ CPU @ 2.70GHz) and 16 GB of memory. The code is implemented in Jupyter-notebook version 3.2.1. All coding is Python 3.9 compliant and in *.ipynb format.

The dataset for experiments is selected from the UCI Machine Learning Repository in [12]. Four known multi-class datasets are used. The definitions of the datasets are shown in Table 1.

Table 1: Features of Used Datasets

Dataset Name	ID	# of Instances	# of Attributes	# of Output Classes
Iris	IRI	150	4	3
Wine	WIN	178	13	3
Maternal Health Risk Data Set	MAT	1014	7	3
Nonverbal tourists	TOU	73	22	6

The important point about the datasets is that 2 of them are old while the final two are recently added to the repository. All of them are classification problems consisting of real, integer, or string values. "Null" values are accepted as 0, and all strings have been converted to their numeric value by a label encoder library.

Each dataset has been divided into training and test datasets with the code stated in Algorithm 1. For any kind of coding process, python has been used, and the Scikit-learn framework in python is an open-source machine learning library that supports supervised and unsupervised learning. It provides even basic and even sophisticated tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities. In other words, scikit-learn provides most of the main features which are needed for a basic working knowledge of machine learning practices (model fitting, predicting, cross-validation, etc.).

Algorithm 1: Train-Test Split

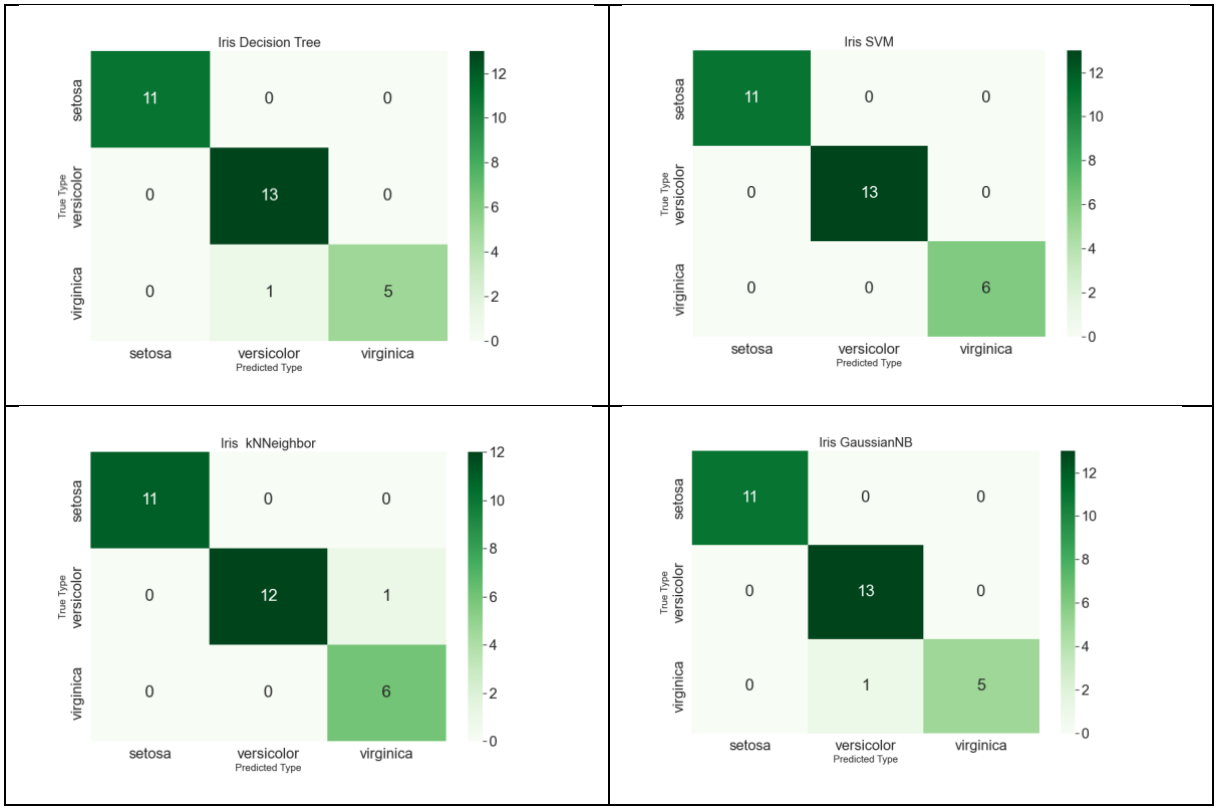
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
                    X, y, test_size=0.2, random_state=42)

```

Then with the help of a related scikit-learn library, all datasets have been split into random train and test subsets. Train dataset contains the random 80% and the test data set contains the rest random 20% of the whole. “Random state” is a seed control value while the shuffling is applied to the data before applying the split.

The related classifiers (Decision Tree, SVM, kNN, and Gaussian NB) use the following parameters because they all have a positive or negative effect on the result. This tuning process is left as a future work for the study.

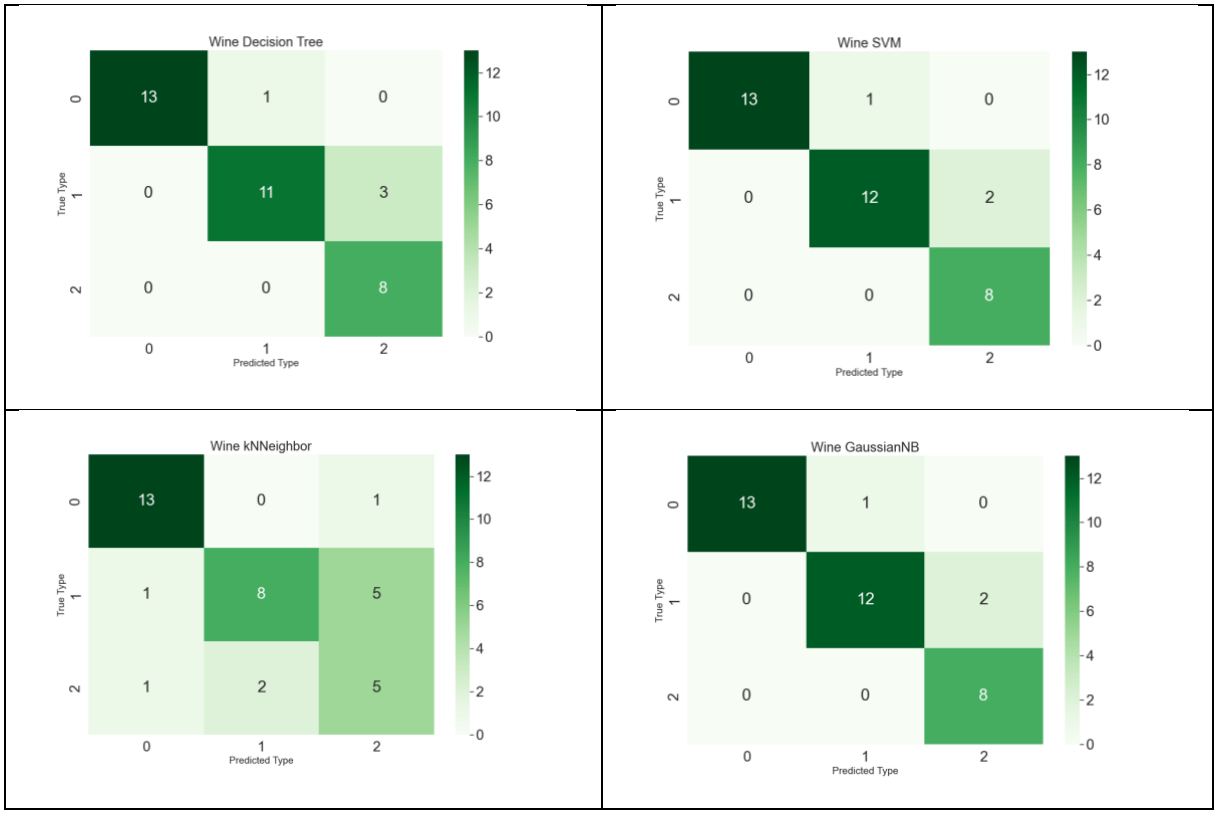
- DecisionTreeClassifier(max_depth = 4)
- SVC(kernel = 'linear', C = 4)
- KNeighborsClassifier(n_neighbors = 5)
- GaussianNB()

Table 2: Confusion Matrix of Classifiers for IRIS Dataset



If to examine the results of each dataset, in Table 2, Iris dataset results are presented using the 4 classifiers stated above

Table 3: Confusion Matrix of Classifiers for WINE Dataset



Especially the results of the final dataset TOU, with 6 output classes worth examining. SVM seems to be the best classifier among others since it has the highest accuracy values in 3 out of 4 datasets. However, the Gaussian NB algorithm has also promised results. If the results are examined due to the size, then it can be inferred that Gaussian NB can be a good alternative.

All results are presented as a Confusion Matrices in Tables 2-5. A confusion matrix can be defined as a summary of prediction results against real values. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the main goal and a brief representation of a confusion matrix.

Besides that, the confusion matrix shows how your classification model is confused when it makes predictions. It provides the insight not only into the errors being made by the selected classifier but more importantly the types of errors that are being made.

Table 4: Confusion Matrix of Classifiers for MAT Dataset

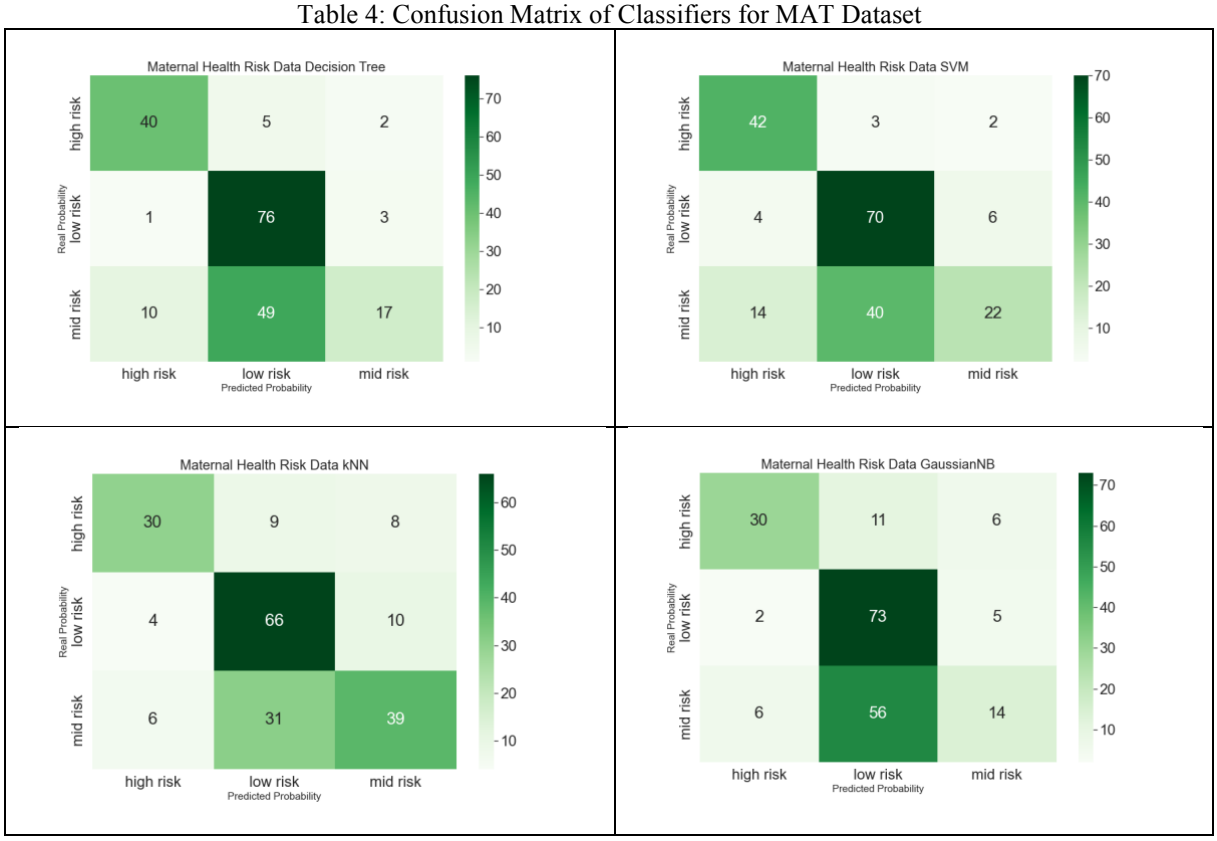
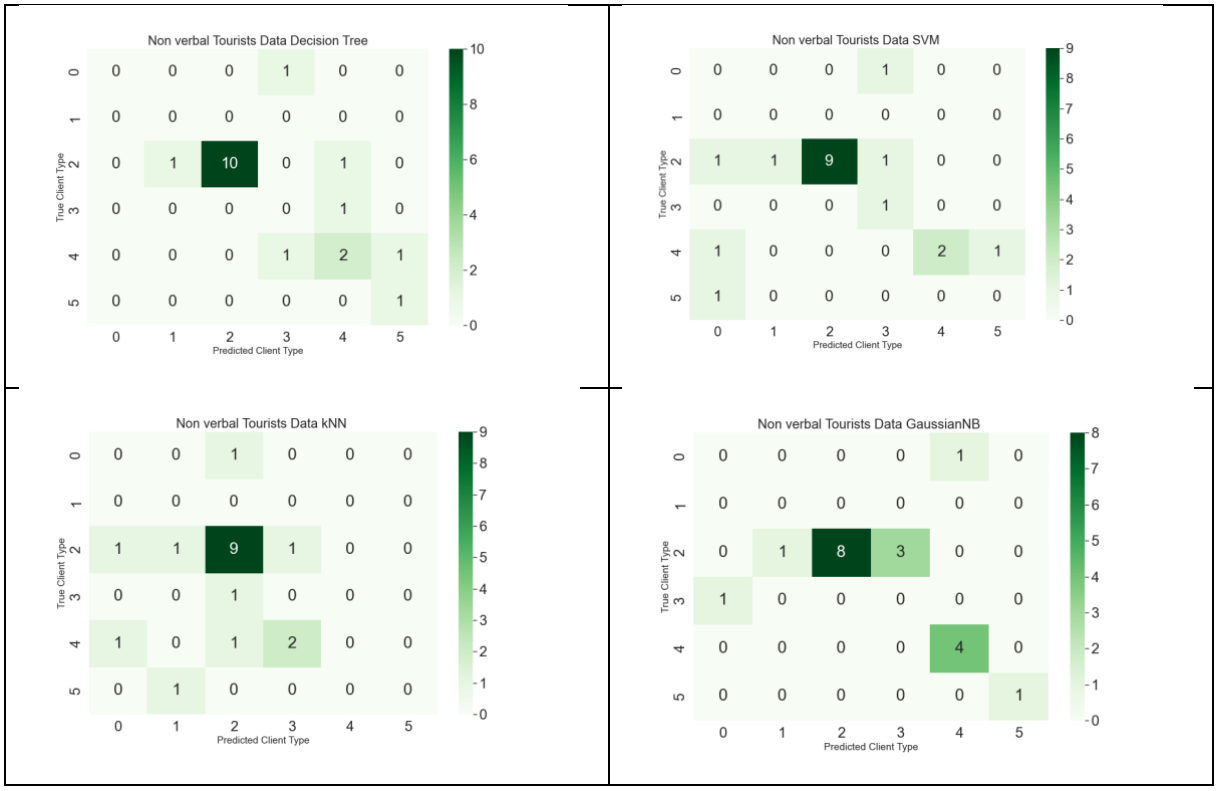


Table 5: Confusion Matrix of Classifiers for TOU Dataset



For example, if we check Table 2 for the Iris dataset solved by Decision Tree, the y-axis shows the true values while the x-axis shows the predicted values. The first row says ("11,0,0") that 11 *setosa* type iris

flower is predicted, which is true, then the second row ("0,13,0") says that 11 *versicolor* type iris flower is predicted, which is true, then finally third row ("0,1,5") says that 5 out of 6 virginica iris flower is predicted, which is true, however, 1 out 6 is predicted as *versicolor*, but it must be *virginica*. The result is we have only one false prediction for this dataset.

When we check the whole accuracy figures, all the results show that SVM is more probable for reaching better results which can be seen in Table 6. The highest values among the classifiers are bolded.

SVM knows the best 3 out of 4, Gaussian NB the best 2 out of 4, and the rest algorithms know the best for only one dataset. This result briefly shows the power of SVM algorithm, esp. for such multiclass datasets.

Table 6: Results for Classifiers

ID	Decision Tree	SVM	kNN Alg.	GaussianNB
IRI	0,9666	**1,0000**	0,9666	0,9666
WIN	0,8888	**0,9167**	0,7222	**0,9167**
MAT	0,6552	**0,6601**	**0,6601**	0,5763
TOU	**0,6842**	0,6316	0,4736	**0,6842**

4. Conclusions and Future Work

In the study, 4 known and accepted algorithms have been executed and solved by using the known and accepted repositories. The results showed that ML algorithms are both easy to use and good at reaching better accuracy in a reasonable time amount. This success can be developed by tuning parameters. Such a capability makes them the best used and preferred methods indisputably among the learning methodologies.

As stated in the previous section, all the experiments must be diversified with different and various types of classifiers. Here, only 4 of them have been used and they will be beneficial for a real-life application. Because as more datasets and characteristics are used, the success rate may degrade or underperform. Different data characteristics, wrong models/classifiers, or lack of time may cause such results.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] Sevinç, Ender. "An empowered AdaBoost algorithm implementation: A COVID-19 dataset study." Computers & Industrial Engineering (2022): 107912.

[2] Mirjalili, Seyedali. "Genetic algorithm." Evolutionary algorithms and neural networks. Springer, Cham, 2019. 43-55.

[3] Karakaya, Murat, and Ender SEVİNÇ. "An efficient genetic algorithm for routing multiple uavs under flight range and service time window constraints." Bilişim Teknolojileri Dergisi 10.1 (2017): 113.

[4] Cingil, Ibrahim, et al. "Dynamic modification of XML documents: External application invocation from XML." ACM SIGecom exchanges 1.1 (2000): 1-6.

[5] Sevinc, Ender. "A novel evolutionary algorithm for data classification problem with extreme learning machines." IEEE Access 7 (2019): 122419-122427.

[6] Dokeroglu, Tansel, and Ender Sevinc. "Memetic Teaching–Learning-Based Optimization algorithms for large graph coloring problems." Engineering Applications of Artificial Intelligence 102 (2021): 104282.

[7] Sevinc, Ender, and Tansel Dokeroglu. "A novel parallel local search algorithm for the maximum vertex weight clique problem in large graphs." Soft Computing 24.5 (2020): 3551-3567.

[8] Too, Jingwei, and Seyedali Mirjalili. "A hyper learning binary dragonfly algorithm for feature selection: A COVID-19 case study." Knowledge-Based Systems 212 (2021): 106553.

[9] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1.1 (1986): 81-106.

[10] Pisner, Derek A., and David M. Schnyer. "Support vector machine." Machine learning. Academic Press, 2020. 101-121.

[11] Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001.

[12] UC Irvine Machine Learning Repository [Online]. Available: https://archive.ics.uci.edu/ml/datasets.php [Accessed: 10-Apr-2022].

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 2, No. 1, July 2022
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi

Reverberation Effect on Online Hazardous Sound Event Detection

Yüksel ARSLAN[1*]
[1]Ankara Science University, Faculty of Engineering and Architecture, Software Engineering Department, Ankara, Türkiye; ORCID: 0000-0003-4791-5534
* Corresponding Author: yuksel.arslan@ankarabilim.edu.tr

Received: 24 April 2022; Accepted: 25 June 2022

Reference/Atıf: Y. Arslan, "Reverberation Effect on Online Hazardous Sound Event Detection" Researcher, vol. 02, no. 01, pp. 29-39, Jul. 2022

Abstract

This paper reports the results of the research on hazardous Sound Event Detection (SED). We used Deep Neural Networks (DNN) to detect car crashes and screams. These are the two of the hazardous sound events on which studies are done for detection. We have selected these sounds because detection of these sounds and early warning can save lives. The research made on hazardous sound events are generally on recorded data. In this paper we wanted to show that there is a difference between recorded data and online (playing) data. At the end if an audio surveillance algorithm would be used in real time, to test it with online data was also an important part of the development. In this research we have developed an online detection environment which consists of a database, automatic audio playing and receiving software, detection software and automatic evaluating software. Our tests show that the reverberation degrades performance significantly. Current research on SED usually only takes into account background noise which is inserted artificially during model development. The results we have found during these online tests are the same as the ones we encountered during far field speaker recognition.

Keywords: audio surveillance, hazardous sound event detection, deep neural networks, reverberation

1. Introduction

Sound event detection (SED) sometimes also called environmental sound recognition (ESR) can be used for many different kinds of purposes. The aim of the SED is to locate temporally and label the sound event classes present in an acoustic signal. For a SED task a set of target sound event classes should be determined. Applications of SED can be listed as follows: There are studies of SED in military, forensic and law enforcement domain. In [1], a gunshot detection system is proposed. In [2], the gunshot blast is used to identify the caliber of the gun. In [3] and [4] SED is used for robot navigation. SED can be used for home monitoring. It can be used to assist elderly people living in their home alone [5],[6]. In [7], it is used for home automation. In [8] and [9], SED is used for recognition of animal sounds. In the surveillance area, it is used for surveillance of road [10], public transport [11], elevator [12] and office corridor [13].

This paper has two contributions to the field: One is to show that the importance of online tests to assess the performance of developed models for SED. The second is the developed online testing environment. Our aim is to develop models that can be used in real environments for real-time audio surveillance. The other studies done before for the acoustic surveillance worked on the recorded data. The studies done previously in this area focus on background noise and signal to noise ratio (SNR). The event sounds to be detected are embedded into different background noises such as traffic, metro station, park etc. at different signal to noise ratio (SNR) levels. In this study during online tests, it is shown that the reverberation is another factor which will affect the performance as much as background noise. It is important to take the reverberation effect during training and testing the model.

In our case, for car crash and scream detection we have prepared event files and background files. The event files are embedded into background files at different SNR levels. Model is developed using these sound files. After model development offline tests are always done as a part of the model development process. If the model reaches required performance during offline tests, then online tests should be

conducted. The same data used in offline case can be played back in laboratory environment to verify that the SED algorithms perform similarly as in offline case [14].

Figure 1: Offline SED [15]

In Figure 1 schematic diagram of offline SED is seen. Audio file consists of the sounds to be detected and the background sounds. In our case the detected sounds are car crashes and screams. The detected sounds are artificially embedded into background sounds. The background sounds are the sounds in which the detected sounds are required to be detected. The audio file is read frame by frame and then features are extracted from these frames. Features are fed into SED algorithm/model for detection. SED algorithm outputs 0 or 1. 0 for normal event and 1 for scream or car crash since we have a model for each detection.

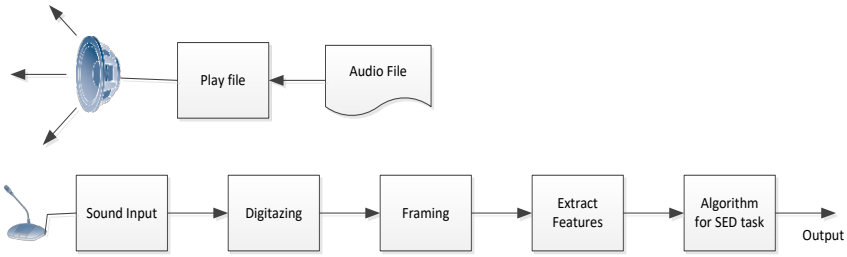


Figure 1: Online SED [15]

In Figure 2 online SED is seen. First the same audio file used in offline SED is read and played from a speaker. A microphone is used to capture these sounds and digitized by the sound card. Digitized input is framed and then same process as in offline case is applied. For online case the frequency range of microphone and speaker is considered. If our features depend on frequency that is the case for most of the time for SED, the frequency characteristics of hardware have big impact on the performance.

Our aim is to find high performance algorithms at the end for online hazardous sound event recognition. For this purpose, we used DNN. Previously other algorithms are used such as Gaussian mixture model (GMM), support vector machine (SVM), hidden markov model (HMM) and other versions of DNN such as convolutional neural networks (CNN) and recurrent neural networks (RNN) on offline data for SED. We followed the usual machine learning methods to find a model for hazardous SED. At last, we used online testing and saw that our quite high-performance model does not work as expected. We can conclude that for real-time applications it is necessary to consider reverberation during development and testing.

The other sections are as follows: In the second section we make a literature review of previous research on the same field. The third section explains the data, feature extraction, model construction for the DNN. In the fourth section we describe the evaluation method and test environment. In the conclusion section we briefly describe our contributions and future work.

2. Literature Review

There has been a lot of research on SED on offline data but has limited number of research on online SED. We have to make a distinction between online and offline research, because all the data is already available in offline case and some computations are possible on the whole data during recognition or in advance. On the other hand, in the online case our algorithms encounter reverberation at different levels depending on the environment and the speed and specs of the devices used must support the algorithms developed.

Recently, the research on SED has been shifted from HMM, GMM and SVM classifiers towards deep learning-based methods such as feed forward neural networks (FNN), CNN, RNN and convolutional recurrent neural networks (CRNN). Detection and Classification of Acoustic Scenes and Events (DCASE) 2017 challenge Task-2 is to detect rare sound events namely glass break, baby cry and gunshot [16]. It is seen on the result page of the challenge that most of the participants have used DNN, CNN or CRNN.

Some papers from the challenge are as follows: A CRNN is used in [17]. In this paper it is declared that convolutional layers of a DNN is capable of learning high-level shift invariant features from time-frequency representations of acoustic samples, while recurrent layers can be used to learn longer term temporal context from the extracted features. These two approaches are combined to detect rare sound events. Among the proposed architectures of CRNN, the best one has an error rate of 0.1733 and F-score 91%.

In [18], it is used 1D convolutional NN as opposed to often used 2D. 128 Mel coefficients are extracted from 46 ms windows of audio signal with 50% overlap. This model achieves 0.1307 error rate and 93% F-score.

In [19], parallel CNN and RNN are used together with Mel coefficients extracted from 40 msec window of input audio signal. This model achieves 0.25 error rate and 86.4 % F-score.

Other research on SED are as follows: For environmental sound classification, deep CNN is used in [20]. This paper explains a classifier based on CNN to classify the environmental sounds such as air conditioner, siren, dog bark and gunshot. This paper also inspects the effect of data augmentation on the performance of CNN. It offers different data augmentation methods to overcome the scarcity of available labelled data. By augmented training data CNN performance increases.

For scream and gunshot detection in noisy environments [21] uses GMM. Two parallel GMM architectures are used for discriminating scream and gunshot from noise. For each classifier, features are selected from a set of 47 features by applying a 2-step selection process.

In [1], a 2-step process is offered to detect gunshot. In the first step an impulsive sound detection process is employed and then a recognition step comes. In recognition step a template matching algorithm with SVM is used.

So far, we mentioned about papers explaining algorithms mainly on recorded data to detect environmental sounds comprising also sounds from hazardous events such as gunshots and screams. From now on we will list some papers which are application oriented, running online or declared in the paper that it is an online algorithm.

In [22], an approach based on the bag of words is proposed for audio surveillance. Authors have prepared a dataset to test the algorithm in realistic complex scenarios. Gunshot, glass break and scream are detected in various background sounds with 6 different SNR values ranging from 5 dB to 30 dB. Recognition rate and false positive rate (FPR) are used as metric. Average recognition rate is 86.7% and FPR is 2.6%.

In [23], the same dataset used in [22] is used to show the performance of a hierarchical RNN. The accuracy of the hierarchical RNN outperforms the work in [22].

In [24], detection, localization and recognition of hazardous sound events are described. This work has a difference from the others such that the embedding of event sounds is not being done on computer. The different environmental noise and the hazardous event recordings are played in an anechoic room and recorded in controlled way. Then the algorithms are run on these recordings.

In [25], audio surveillance is used for car crash detection. The audio signals are divided into short time frames and then features such as spectral spread, volume, energy, zero crossing, energy in 4 sub-bands etc. are extracted. For M classes to be detected M + 1 SVM are used to detect these sound classes and the background sound. This paper also discusses the architectural deployment of such a system in real environments.

3. Model Preparation

SED consists of two stages. First stage is to represent the sound and the second stage is classification. To represent sound, fixed length frames of the sound are taken and some features are extracted from these frames. First sound signals are divided into fixed length frames and then these frames are divided into windows. Then MFCC features are extracted from windows of a frame. The feature extraction can be defined formally as follows: x is the vector of acoustic features obtained from one frame of sound signal. x is obtained from matrix M with $\mathbb{R}^{SxF}$, where $S \in N$ is the number of features per window and $F \in \mathbb{N}$ is the total number of windows per frame. Then x is the vectorization of the matrix M. M is vectorized by concatenating each column. (Figure **2**) The DNN (classifier) task is to find frame probability $\hat{y} = p(y \mid x,\theta)$ for target output $y \in \{0,1\}$(One class output), where $\hat{y}$ denotes the probability of target event in the frame and θ is the parameters of the classifier. Then by applying a threshold on the frame probability, the estimated event z found, if $z = 1$ the event is present in the frame or if $z = 0$ the event is not present [17]. The parameters θ are trained by supervised learning.

The parameters (θ) of the classifier are found through training of the classifier by giving the labelled features extracted from the frames of the training data. This is called supervised learning. In the following section we explained how we found the data and its details.

Car crash dataset (Table 1) is taken from the research described in [25]. This dataset contains 56 files, in four folds. The duration of each of these files are 3 min. We have used three folds for training and last fold for testing. The dataset we have used totally contains 204 car crashes. These are inserted in 56 background files at 15 dB SNR level. Audio files are sampled at 32 KHz. The detailed explanation of the dataset can be found in [25].

Table 1: Car Crash Dataset [15]

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Car crash	150 (events)	54 (events)
Traffic	42	14

Scream dataset is taken from the research explained in [22]. The scream dataset contains 66 files for training and 20 files for testing. Each file has a 3 min duration. The dataset contains sound files at 6 different SNR values, but we have taken only the files with SNR values of 15 dB. Table 2 shows the scream dataset properties. The 2084 scream even sounds are inserted 86 background files. The background files contain different sounds such as metro, park, traffic e.g.

Table 2: Scream Dataset [15]

Training Dataset		**Test Dataset**
Type	**Number of Files**	**Number of Files**
Scream	1881(events)	203 (events)
Various	66	20

Figure 3 shows the feature extraction process that is used in detection of scream, and car crash. Each sound signal is divided into frames according to the predetermined length. These frames are then divided into 40 msec windows with 50% overlapping. From each 40 msec window, 40 MFCCs are computed. Then concatenating all MFCCs of one frame, we obtain the feature vector of the frame. So, each event sound has different size feature vector. These vectors have the output label 1. The same procedure with the same framing and windowing sizes are applied to background sounds. The obtained vectors are labelled as 0.

The minimum length of event signals and the number of MFCCs which are contained in their feature vector is shown in Table 3.

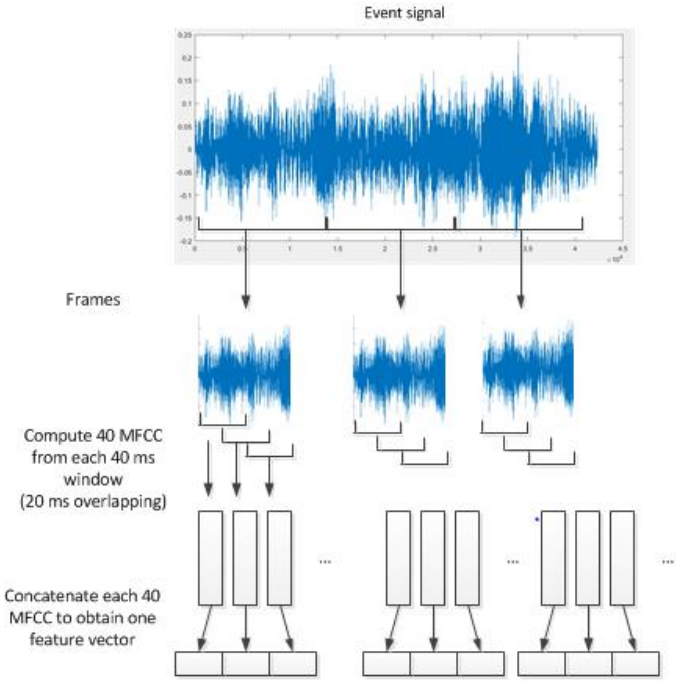


Figure 2: Feature Extraction [15]

Table 3: Event Sound Feature Properties [15]

Event Name	**Minimum Duration (ms)**	**Feature vector MFCC count**
Car crash	711	1360
Scream	490	920

We used DNN as classification algorithm to recognize the sound events. Two DNN models have been developed for each event type. DNN is a supervised and parameterized learning method. In supervised learning, we are given a set of input–target output pairs, and the aim is to learn a general model that maps the inputs to target outputs. Supervised learning methods aim to learn a model that can map the inputs to their target outputs for a given set of training examples. During model generation the model is also tested by using examples not used during learning. Table 4 summarizes the hyper parameters of the DNN used. We used rectified linear unit (RelU) as activation functions of hidden units and sigmoid function at the output unit. Figure 4 shows the DNN architecture used for car crash recognition.

Table 4: DNN Hyper-parameters [15]

Hyper - parameters	**Car crash**	**Scream**
# of layers	4	3
Learning rate	0.085	0.075
Number of iterations	1000	500
Hidden Unit Activation	RelU	RelU
Output activation function	Sigmoid	Sigmoid

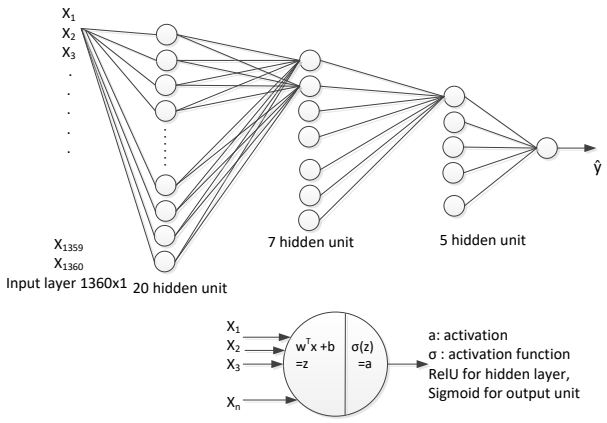


Figure 3: DNN Architecture for Car Crash Detection [15]

We have found hyper-parameters manually by making extensive tests. Grid search algorithms can be used to find best hyper-parameters automatically. Our aim is to find hyper-parameters for an acceptable performance and make offline and online testing with these parameters.

4. Testing

If we will use SED for audio surveillance, the developed algorithms or models will be used eventually in real life applications. The models we developed for road, home or elevator surveillance eventually will be installed in real life environments. After model development and testing with offline data it is necessary to see the efficiency of the models by switching more realistic scenarios.

4.1 Offline Tests

Offline testing provides fast testing on large datasets while online data speed is the recording length of the dataset. Although offline testing is not realistic, it provides us to test our system with large data sets in shorter time.

We read sound files containing the hazardous sound event embedded in a background noise in frames of minimum event length. From these frames we calculate MFCC features as in the training of the model. Then the obtained features are fed into DNN model. These tests are done for each event type separately. A sound file may contain scream and car crash events at the same time, but we detect just scream in scream model tests.

F-score, error rate and other metrics such as accuracy, true positive and false positive rates are used for model evaluation. In SED it is better to check more than one metric at the same time to evaluate the model performance. In SED surveillance applications the most important metrics are recognition rate, true positives, false positives and error rate. False positives and error rate are very important in real applications because the user can stop using application if he/she get many false warnings. The definitions for the metrics and formulas are as follows [27]:

True positives (TP): an event in the system output that has a temporal position overlapping with the temporal position of an event with the same label in the reference. A collar is usually allowed for the onset and offset, or a tolerance with respect to the reference event duration.

False positives (FP): an event in the system output that has no correspondence to an event with same label in the reference within the allowed tolerance.

False negative (FN): an event in the reference that has no correspondence to an event with same label in the system output within the allowed tolerance.

True negative (TN): truly not detected events.

Insertions (I): the number of events in system output that are not correct

Deletions (D): the number of reference events that were not correctly identified

Substitutions (S): events in the system output that have correct temporal position but incorrect class label.

Total events (N): total number of events need to be detected.

$$\text{Error rate (ER)} = \frac{S+I+D}{N} \quad (1)$$

$$\text{F-score} = \frac{2 .P. R}{P+R} \quad (2)$$

$$\text{P} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{R} = \frac{TP}{TP+FN} \quad (4)$$

The metrics defined in [27] are for polyphonic sound event detection, in this paper we deal with monophonic sound event detection where each sound clip contains one type of event sounds. For our case we can write the error rate as follows:

$$\text{Error rate (ER)} = \frac{I+D}{N} = \frac{FP+FN}{N} \quad (5)$$

TP rate which is also called sensitivity is the percentage of events correctly detected. FP rate is percentage of non-event frames labelled as an event. Recognition rate is the percentage of TPs, TNs to the total frames such that TPs, TNs, FPs and FNs. The formulas are as follows:

$$\text{TPR (sensitivity)} = \frac{TP}{TP+FN} \quad (6)$$

$$\text{FPR} = \frac{FP}{FP+TN} \quad (7)$$

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

These metrics are explained in more detail in [27]. As in [16] 500 msec tolerance was used for detection of events in this work. For offline detection, we assume the detected event on time if the detected time is 500 msec before the event start time or 500 msec after event end time. For online detection 1 sec tolerance was used. The performance of scream detection is seen in Table 5. Table 6 shows comparison results of scream detection with the results of the research in [22] and [23]. The proposed method in this work has a recognition rate 98.4% and outperforms [22] which is 87% and it is very close to the work done in [23].

Table 5: Performance of Scream Detection [15]

Accuracy (%)	**TPR (%)**	**Error rate**	**F-score (%)**
98.4	87.6	0.47	75.4

Table 6: Accuracy Comparison of Scream Detection with Proposed Method and Other Two Studies on The Same Dataset [15]

Proposed method (%)	**Foggia et al. (%)**	**Colangelo et al. (%)**
98.4	87	98.5

Car crash detection results are seen in Table 7.

Table 7: Car Crash Detection Performance [15]

Accuracy (%)	**TPR (%)**	**Error rate**	**F-score (%)**
98.4	77.7	0.35	81.5

The comparison of the car crash results with the work in [25] are seen in Table 8.

Table 8: Accuracy Comparison of Car Crash Detection with Proposed Method and The Work in [25] on The Same Dataset [15]

Proposed method (%)	**Foggia et al. [25] (%)**
98.4	84.5

4.2 Online Tests

For online tests an automatic testing environment was prepared as shown in Figure 5. The microphone and speaker frequency responses are important for the tests to be successful. The audio clips used in offline tests are sampled at 32 KHz for scream and car crash which means the sounds can have frequency components at most at 16 KHz. Therefore, the microphone and the speaker we used must support at least 16 KHz frequency as we used MFCC as feature and it depends on the audio clips frequencies. For this reason, a microphone and speakers which supports the frequency range from 80 Hz to 20 kHz are selected in online testing environment.

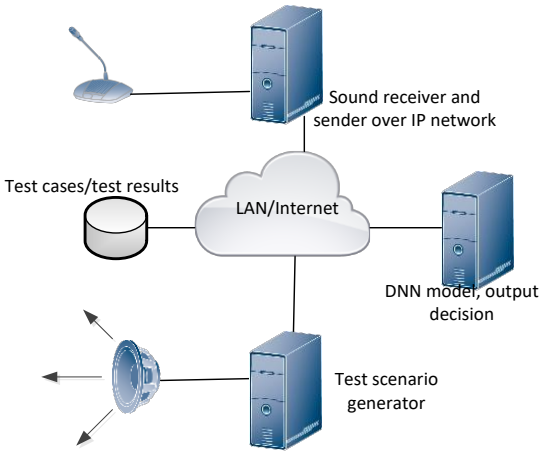


Figure 4: Online Test System [15]

We have prepared and installed three software on the computers shown in Figure 5 which we call online test system.

DNN model/output decision: This software runs the car crash and scream models. It has a user datagram protocol (UDP) receiver, and it takes the sound packets over network. We developed it as a standalone server that can run in distant place from the sound source, it can be a computer over cloud.

Test scenario generator/player: It reads the test sound files, writes the start time of playing of the sound files as offset and the start time of the events to the test cases table of the database. Then it plays each test sound file one by one.

Sound receiver and sender over IP network: The third software captures the played sound with a microphone and sends it over the local area network (LAN) or over Internet using UDP to the recognition software, namely it is DNN model/output decision software. If the recognition software detects car crash or scream it writes its decisions and the detection time to the test results table of the database. After inspecting test cases and test results the performance of the online system is found.

The online tests were done in three different room environments and in anechoic room. When the microphone is close to speakers such that the distance was less than 30 cm, the performance was the same as offline tests. In all three rooms we achieved almost the same results as offline case. When the distance is greater than 50 cm, the performance of online tests was degraded significantly. We repeat the same online tests in an anechoic room. We set the distance 2 m in this anechoic room, and we obtained the same results as offline tests.

Table 9 and Table 10 show the online and offline test result comparisons of scream and car crash detection respectively. Online results shown in tables below are the results obtained when the microphone to speaker distance is 30 cm.

Table 9: Comparison of Offline and Online Scream Detection [15]

	TPR	Error rate	F-score(%)
Offline	87.6	0.47	75.4
Online	80.2	0.50	72.3

Table 10: Comparison of Offline and Online Car Crash Detection [15]

	TPR	Error rate	F-score(%)
Offline	77.7	0.35	81.5
Online	75.4	0.40	75.7

4.3 Discussion

Online tests show that the performance of offline tests can only be achieved if the microphone is close to speaker. In our case, in approximately 30 cm we obtained the same results. If we increase the distance, performance will decrease gradually, it is half of the offline performance at about 50 cm. This performance decrease is due to reverberation. To prove this, we repeat the online tests in an anechoic room. In the anechoic room, we could place the microphones 2 m apart at most because the anechoic room that we could access was a small one. In this anechoic room, we measured the performance the same as offline performance.

The effect of of distance in speaker recognition is a known and studied problem. We encountered the same problem when we are making hazardous SED tests online. The problem of distant speaker recognition (DSR) can be explained as whenever speaker to microphone distance increases, recognition rates decrease and equal error rate (EER) increase [28]. In speaker recognition solutions such as reverberation compensation, feature warping or using multiple microphones can improve performance significantly [29].

In online hazardous SED we can use the followings to improve the performance:

- During training and model preparation we can prepare a sound database degraded with reverberation manually.
- We record pure event sounds and then degrade it with different level of reverberation and then use these sounds for training.
- We record pure event sounds and play and record the these sounds just at the same place where this SED application will be used. We then use these sounds for training and model preparation.
- We can develop the model as we did before but we we can apply de-reverberation methods before giving the sound to DNN model.

5. Conclusion

In this work we showed that hazardous SED models developed will show poor performance in real world applications. To use these models in a real-world scenario reverberation should be reconsidered. To show this after developing car crash and scream models we tested them online. The online performance of these models is less than offline performance when the microphone is apart from speaker. We proved that the reason is reverberation by repeating the tests in anechoic room.

We propose if the SED is used for an online application reverberation must be considered. As the future works the proposed de-reverberation techniques can be applied during online tests. The proposed four methods can be used to remove the reverberation effect and other methods can be proposed. Finally, the best one or combination can be used.

Conflicts of Interest

The author declares no conflict of interest.

References

[1] T. Ahmed, M. Uppal and A. Muhammad, “Improving Efficiency and Realibility of Gunshot Detection Systems”, IEEE, ICASSP 2013.

[2] P. Thumwarin, T. Matsuura and K. Yakoompai, "Audio forensics from gunshot for firearm identification", Proc. IEEE 4th Joint International Conference on Information and Communication Technology Electronic and Electrical Engineering Tailand, pp. 1-4, 2014.

[3] S. Chu, S. Narayanan, C.J. Kuo, M.J. Mataric,., “Where am I? Scene recognition for mobile robots using audio features”, in 2006 IEEE Int.Conf. on Multimedia and Expo. IEEE, 885–888, 2006.

[4] N. Yamakawa, T. Takahashi, T. Kitahara, T. Ogata, H.G. Okuno, “Environmental sound recognition for robot audition using Matching-Pursuit”, International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer Berlin Heidelberg, 1–10, 2011.

[5] J. Chen, A.H. Kam, J. Zhang, N. Liu, L. Shue, “Bathroom activity monitoring based on sound”, in Pervasive Computing, Springer Berlin Heidelberg, 47–61, 2005.

[6] M. Vacher, F. Portet, A. Fleury, N. Noury, “Challenges in the processing of audio channels for ambient assisted living”, in 2010 12th IEEE Int. Conf. on e-Health Networking Applications and Services (Healthcom), IEEE, 330–337, 2010.

[7] J.C. Wang, H.P. Lee, J.F. Wang, C.B. Lin, “Robust environmental sound recognition for home automation”., Automation Science and Engineering, IEEE Transactions on, 5 (1) (2008), 25–31.

[8] R. Bardeli, D. Wolff,, F. Kurth, M. Koch, K.H. Tauchert, K.H. Frommolt, “ Detecting bird sounds in a complex acoustic environment and application to bioacoustic monitoring”, Pattern Recognition. Letters, 31 (12) (2010), 1524–1534.

[9] F. Weninger, B. Schuller, “Audio recognition in the wild: static and dynamic classification on a real-world database of animal vocalizations”, in 2011 IEEE Int.Conf. on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, 337–340.

[10] P. Foggia, A. Saggese, N. Strisciuglio, M. Vento, and N. Petkov, “Car crashes detection by audio analysis in crowded roads”, In Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on, pages 1-6, Aug 2015.

[11] J. Rouas, J. Louradour, and S. Ambellouis, "Audio Events Detection in Public Transport Vehicle," Proc. of the 9th International IEEE Conference on Intelligent Transportation Systems, 2006.

[12] R. Radhakrishnan and A. Divakaran, “Systematic acquisition of audio classes for elevator surveillance,” in Image and Video Communications and Processing 2005, vol. 5685 of Proceedings of SPIE, pp. 64–71, March 2005.

[13] P. K. Atrey, N. C. Maddage, and M. S. Kankanhalli, “Audio based event detection for multimedia surveillance,” in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’06), vol. 5, pp. 813–816, Toulouse, France, May 2006.

[14] T. Virtanen, M. Plumbley, D. Ellis, “Computational Analysis of Sound Scenes and Events”, book, Springer, 21 Sep. 2017.

[15] Arslan Y. Detection and recognition of sounds from hazardous events for surveillance applications. PhD, Yıldırım Beyazıt University, Ankara, Turkey, 2018

[16] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, “DCASE 2017 challenge setup: tasks, datasets and baseline system”, in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2017 Workshop (DCASE2017), November 2017.

[17] E.Cakır and T. Virtanen, “Convolutional Recurrent Neural Networks for Rare Sound Event Detection”, DCASE 2017, 27 Nov. 2017.

[18] H. Lim, J. Park, K. Lee, Y.Han, “Rare Sound Event Detection Using 1D Convolutional Recurrent Neural Networks ”, DCASE 2017, 27 Nov. 2017.

[19] A. Dang, T. H. Vu, J. C. Wang, “Deep Learning For DCASE 2017 Challenge”, DCASE 2017, 16 Nov. 2017.

[20] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification”, IEEE Signal Processing Letters, Vol. 24, No.3, March 2017.

[21] L. Gerosa, G. Valenzise, F. Antonacci, M. Tagliasacchi and A. Sarti, "Scream and gunshot detection in noisy environments," in EURASIP, Poznan, Poland, September 2007.

[22] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, M. Vento, "Reliable detection of audio events in highly noisy environments", Pattern Recognition Letters, vol. 65, pp. 22-28, 2015.

[23] F. Colangelo, F. Battisti, M. Carli, A. Neri, “Enhancing audio surveillance with hierarchical recurrent neural networks”, Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on, Sept 2017.

[24] K. Lopatka J. Kotus, A. Czyzewski, “Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations”, Multimedia Tools and Applications, 75:1–33, 2016.

[25] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds", IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 1, pp. 279-288, Jan 2016.

[26] Y.Arslan and H. Canbolat, "A sound database development for environmental sound recognition", Signal Processing and Communications Applications Conference (SIU), 25th, 2017.

[27] A. Mesaros, T. Heittola, T. Virtanen, "Metrics for Polyphonic Sound Event Detection" Applied Sciences, vol. 6, no. 6, p. 162, 2016.

[28] M. A. Nematollahi, S. A. R. Al-Haddad, "Distant speaker recognition: an overview", International Journal of Humanoid Robotics, pp. 1550032, 2015.

[29] Q. Jin, T. Schultz, A. Waibel, "Far-Field Speaker Recognition", *IEEE TASLP*, vol. 15, no. 7, pp. 2023-2032, Sept. 2007.

ANKARA SCIENCE UNIVERSITY, RESEARCHER
Vol. 2, No. 1, July 2022
e-ISSN: 2717-9494
Research Article/Araştırma Makalesi

Big Data Reduction and Visualization Using the K-Means Algorithm

Hakan AKYOL[1], Hale Sema KIZILDUMAN[2], Tansel DÖKEROĞLU[3*]
[1]Çankaya University, Graduate School of Natural and Applied Sciences, Ankara, Türkiye; ORCID: 0000-0002-5695-8790
[2]Çankaya University, Graduate School of Natural and Applied Sciences, Ankara, Türkiye; ORCID: 0000-0002-6449-771X
[3]Çankaya University, Software Engineering Department, Ankara, Türkiye; ORCID: 0000-0003-1665-5928
*Corresponding Author: tdokeroglu@cankaya.edu.tr

Received: 25 June 2022; Accepted: 30 June 2022

Reference/Atıf: H. Akyol, H. S. Kızılduman and T. Dökeroğlu, "Big Data Reduction and Visualization Using the K-Means Algorithm" Researcher, vol. 02, no. 01, pp. 40-45, Jul. 2022

Abstract

A huge amount of data is being produced every day in our era. In addition to high-performance processing approaches, efficiently visualizing this quantity of data (up to Terabytes) remains a major difficulty. In this study, we use the well-known clustering method *K*-means as a data reduction strategy that keeps the visual quality of the provided huge data as high as possible. The centroids of the dataset are used to display the distribution properties of data in a straightforward manner. Our data comes from a recent Kaggle big data set (Click Through Rate), and it is displayed using Box plots on reduced datasets, compared to the original plots. It is discovered that *K*-means is an effective strategy for reducing the amount of huge data in order to view the original data without sacrificing its distribution information quality.

Keywords: big data, data reduction, visualization, *k*-means

1. Introduction

Data visualization is the way of representing your data using graphical/visual elements to perceive and analyze your data in shorter times and more meaningfully [1]. By utilizing visual components such as charts and graphs, data visualization tools ease to identify and analyze trends, outliers, and patterns in data. However, big data analytics come with new problems and research opportunities for the visualization of the data [2]. Dealing with large volumes of data is far more difficult than dealing with small amounts of data [3]. Enrico and Antonio present a detailed survey about the recent developments and research areas of big data analytics and visualization in their study. Studies in this area have still been continuing [4][5].

In this study, we maintain the visual quality of the box plots (which give information about the distribution of the data) while reducing the size of big data. During this study, we clustered the data with the *K*-means algorithm and used the obtained centroids in our visual elements [6]. Thus, we obtain similar plots with fewer data while keeping the data distribution information of the big data [7].

2. Data Reduction Techniques

This section briefly explains the data reduction techniques we have used in our study.

Randomized Data Reduction: We employ the randomized data reduction approach to compare the performance of the results obtained using the *K*-means algorithm. *n* many data instances are chosen at random from the large data collection, and graphs are drawn using this data. During this procedure, no sampling approach is employed. This serves as a benchmark for evaluating the quality of our *K*-means algorithm outcomes. In order to be fair with our comparisons, we take the same size random values and *K* values.

Data Reduction using the *K*-means clustering algorithm: The algorithm aims to divide data instances into *K* clusters, with each trial belonging to the cluster with the cluster centroid. *K*-means clustering minimizes within-cluster variances. This technique is computationally hard. However, heuristic

algorithms can quickly report near-optimal solutions easily. Therefore, it can be used to select the most representative data instances to give information about the distribution of the big datasets.

3. Experimental setup and evaluation of the results

The datasets we have used in our experiments are Click Through Rate (CTR) from Kaggle [8]. The prediction of advertisement CTR is an important challenge in the field of computational advertising. Increasing the accuracy of advertising CTR prediction is crucial for improving precision marketing efficacy. The dataset discloses large anonymised advertising datasets. There are one million instances in this big dataset.

The visual elements in our study are produced with a PC having an i7 processor, 16 GB RAM, 64-bit operating system, and 8 GB Intel(R) HD Graphics 630 + 4GB NVIDIA GeForce GTX 1050 graphics card. Pycharm IDE is used. The python version is Python 3.9.12. The packages used are pandas, numpy, statsmodels.api, matplotlib.pyplot, seaborn and sklearn.cluster *K*-Means.

In Figure 1, the visualization of the *city* column of the dataset (with 10, 100, 1000, 10000, and original data sizes) are presented. The data is obtained from the first rows of the original dataset. As the selected datasets get bigger, they represent the distribution of the original dataset in a better way. Dataset with 10 instances has the biggest deviation in terms of median values from the original dataset's median. As it can be seen the lowest and highest values of the data are different from the original dataset. In Figure 2, we give the visualization of K-mean results (centroids) with 10, 100, 200, and 500 (this was the biggest *K* value we can get during our experiments) size datasets. The distribution of the datasets is matched. Because the number of instances is very few, all the data cannot be seen in the plot. However, with *K*=500, a plot very similar to the original data is obtained.

Figure 3 gives the distribution of the *city* and *device_size* columns of the dataset (with randomly selected 100, 1000, 10000, and original data sizes). Figure 4 gives the data distribution of the *city* and *device_size* column of the dataset by producing the data with the *K*-means algorithm using *K*=10, *K*=100, *K*=200, and *K*=500. Although the frequency of the data cannot be seen in Figure 4, a better visualization than selecting random slices of data is provided. As the value of K increases, better plots are available. Figures 5 and 6 present the *gender* and *device_size* data visualizations from our dataset in the same way and the reader can easily see the higher quality of the plots with the results of *K*-means. The plot with *K*=500 is almost the same as the original dataset's plot whereas *K*=10 cannot match the upper values of the *Gender* axis. Approximately using 0.0005% of the original dataset, we have drawn clear plots of the big data with the *K*-means algorithm. The execution time of the *K*-Means algorithm is reasonable up to 100 centroids.

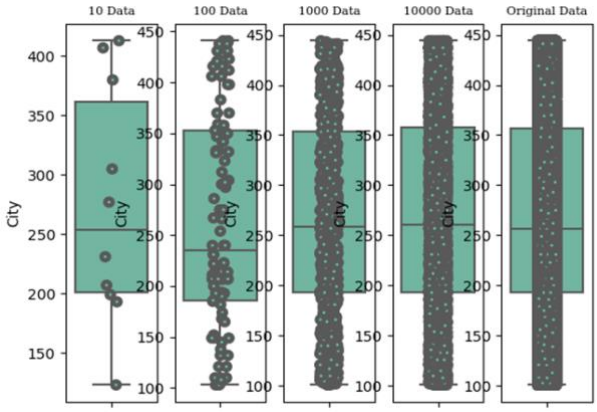


Figure 1: The Distribution Visualization of The *City* Data of The Dataset (with randomly selected 10, 100, 1000, 10000, and original data sizes).

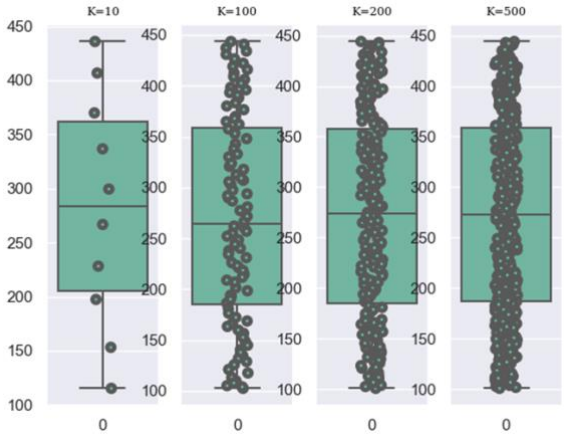


Figure 2: The Distribution of The *City* Data of The Dataset by Producing the Data with the *K*-Means Algorithm. *K*=10, K=100, *K*=200, and *K*=500 are presented in the respective columns.

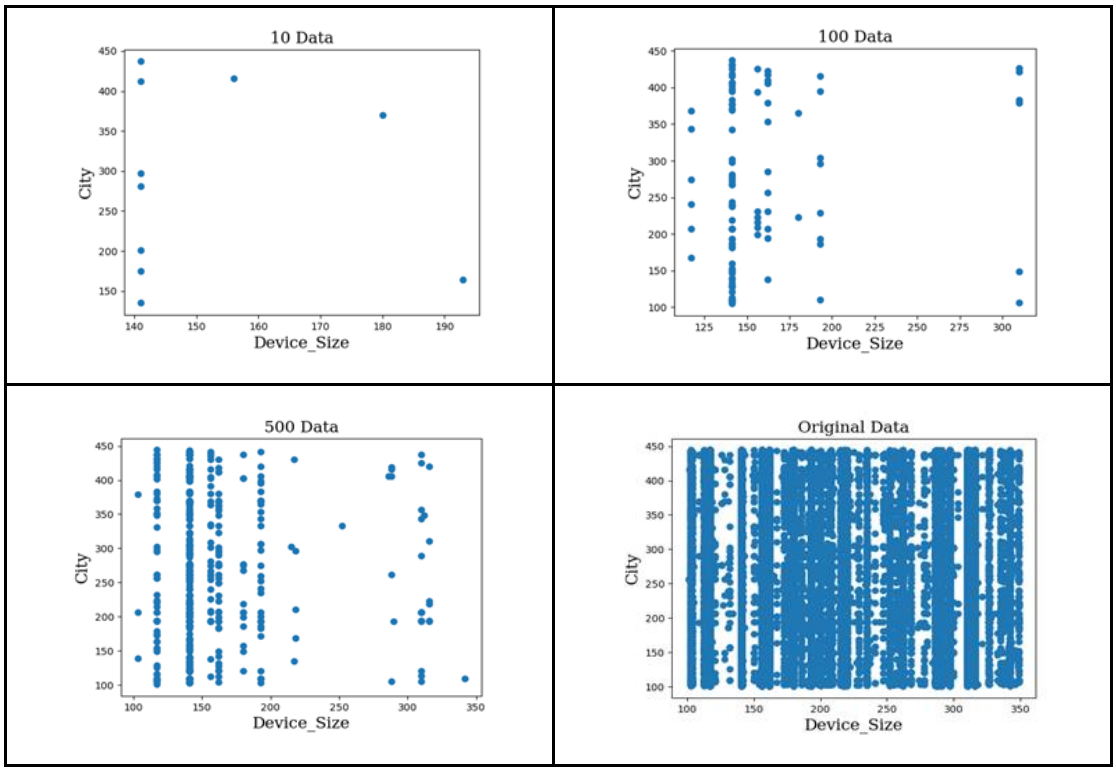


Figure 3: The Data Distribution of The *City* and *Device_Size* Data of The Dataset (with randomly selected 100, 1000, 10000, and original data sizes).

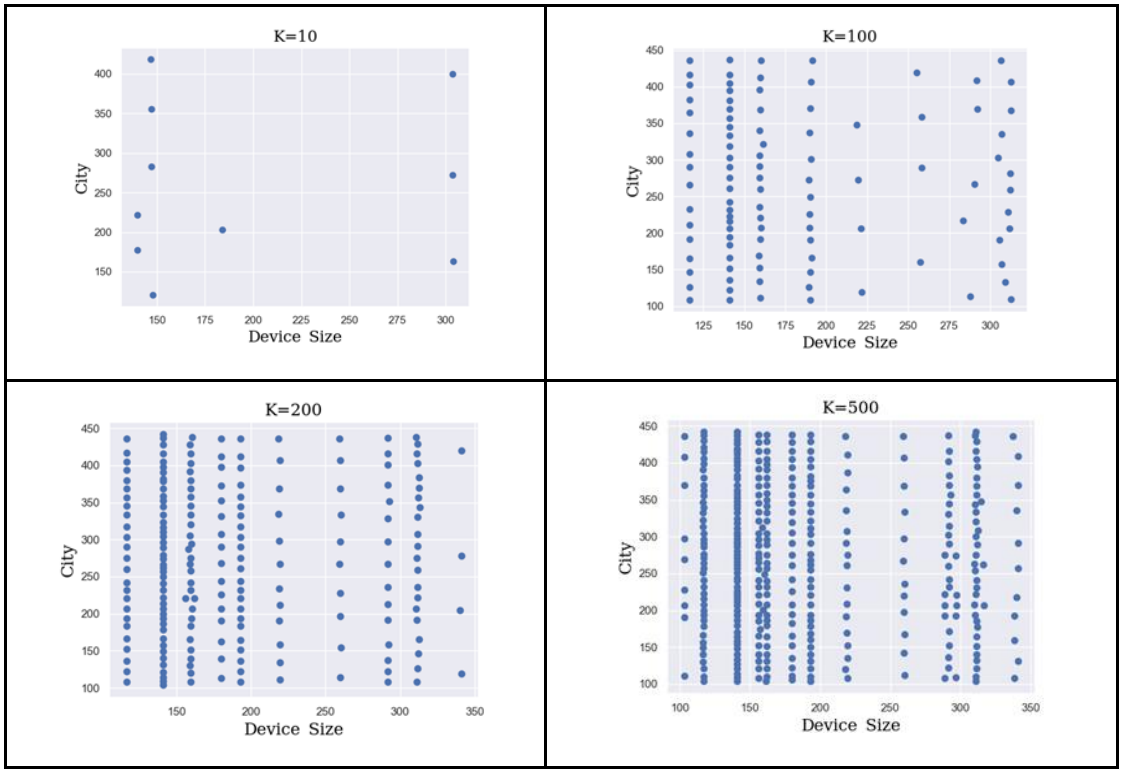


Figure 4: The Data Distribution of The City and *Device_Size* Data of The Dataset by Producing the Data with the *K*-Means Algorithm. *K*=10, *K*=100, *K*=200, and *K*=500 are presented in the respective columns.

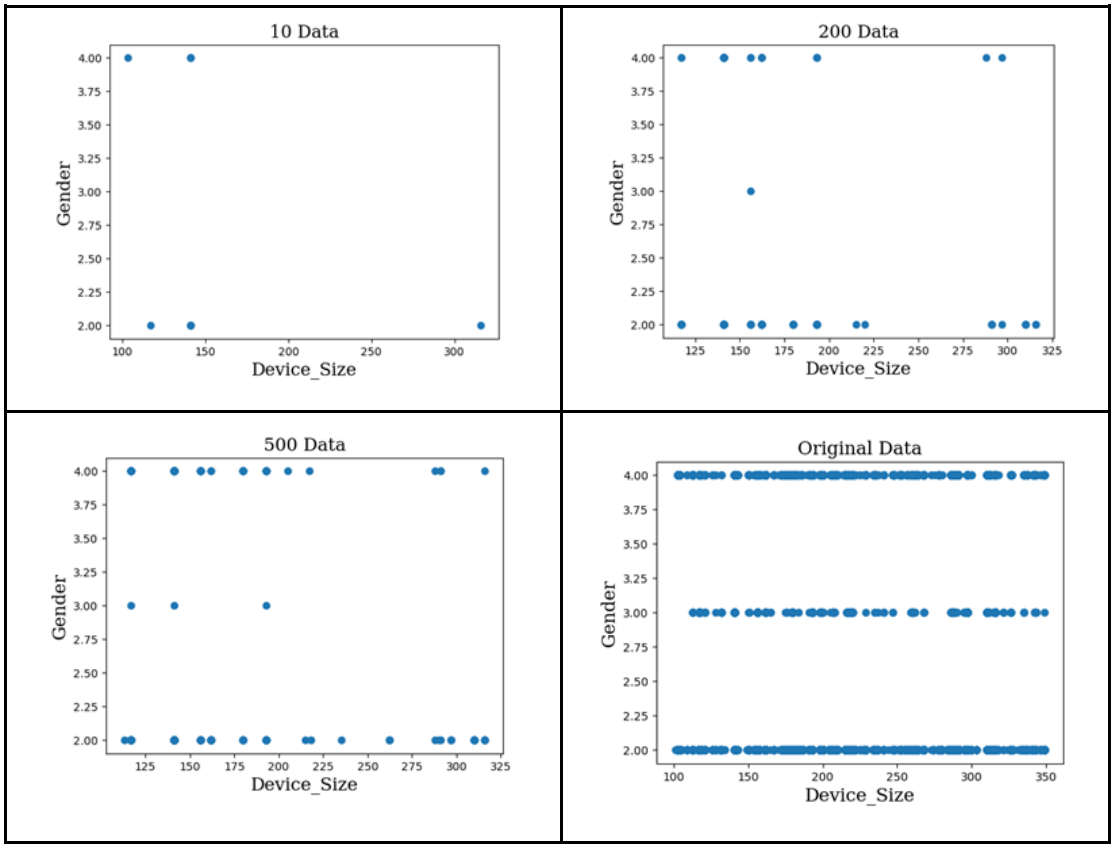


Figure 5: The Data Distribution of The *Gender* and *Device_Size* Data of the Dataset (with randomly selected 10, 200, 500, and original data sizes).

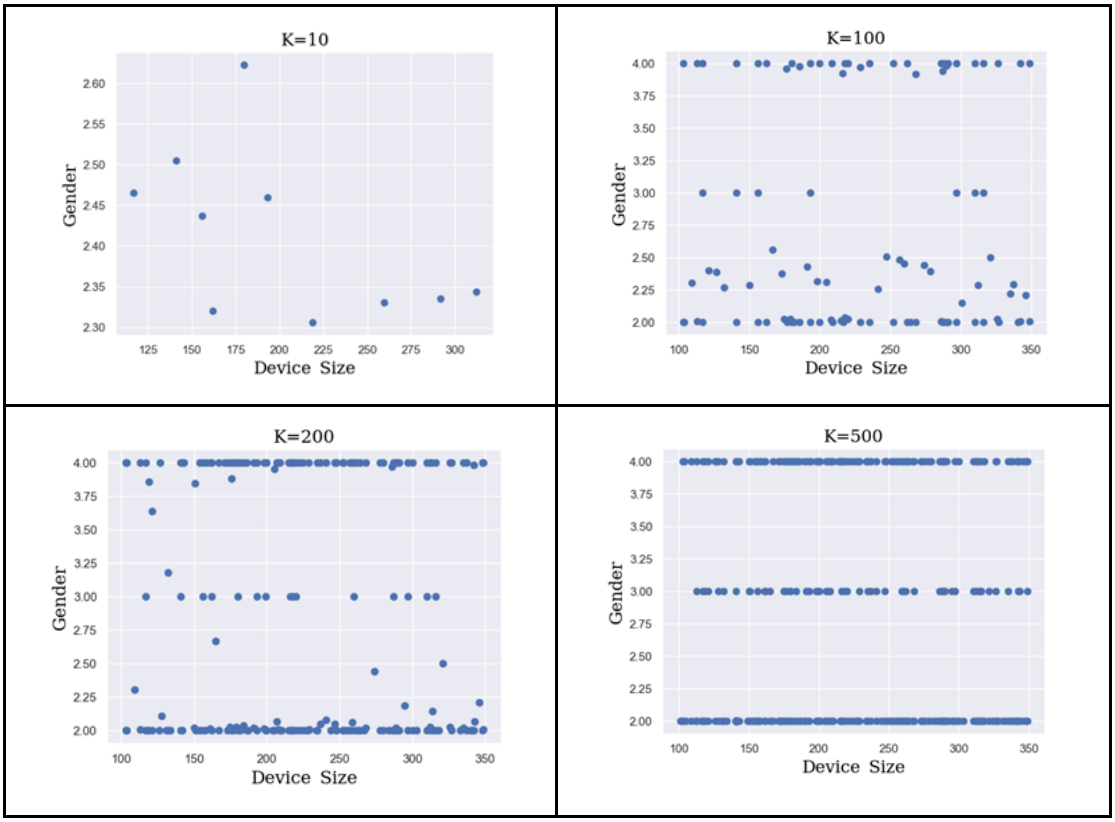


Figure 6: The Data Distribution of The *Gender* and *Device_Size* Data of The Dataset by Producing the Data with the *K*-Means Algorithm. *K*=10, *K*=100, *K*=200, and *K*=500 are presented in the respective columns.

4. Conclusion and future work

Although *K*-means is a clustering algorithm to set the best set of centroids, in this study, it is used as a technique to select/reduce the most indicative data instances so as to visualize big data sets. From the results of our dataset, we have observed the distribution information of the dataset can be kept with a smaller set of data instances obtained as centroids using the *K*-means algorithm. This visualization problem is still a hot topic for researchers. According to the behavior of the datasets, visualization will always be a critical issue for decision-makers. To the best of our knowledge, the method we propose here is the first application of the *K*-means algorithm to the visualization of big data to represent the original data with a reduced set.

In our future work, we intend to study with much bigger datasets and use a big data visualization tool such as Tableau, QlikView, or Microsoft Power BI. We will compare the visual elements of the reduced sets of original big data sets and try to keep the quality and informative features of the data as high as possible.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Friendly, M. (2008). A brief history of data visualization. In Handbook of data visualization (pp. 15-56). Springer, Berlin, Heidelberg.

[2] Keim, D., Qu, H., & Ma, K. L. (2013). Big-data visualization. IEEE Computer Graphics and Applications, 33(4), 20-21.

[3] Andrienko, G., Andrienko, N., Drucker, S., Fekete, J. D., Fisher, D., Idreos, S., ... & Sharaf, M. (2020, March). Big data visualization and analytics: Future research challenges and emerging applications. In BigVis 2020-3rd International Workshop on Big Data Visual Exploration and Analytics.

[4] Agrawal, R., Kadadi, A., Dai, X., & Andres, F. (2015). Challenges and opportunities with big data visualization. In Proceedings of the 7th International Conference on Management of computational and collective intElligence in Digital EcoSystems (pp. 169-173).

[5] Ali, S. M., Gupta, N., Nayak, G. K., & Lenka, R. K. (2016). Big data visualization: Tools and challenges. In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) (pp. 656-660). IEEE.

[6] Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. Pattern recognition, 36(2), 451-461.

[7] Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2022). A Comprehensive Survey on Recent Metaheuristics for Feature Selection. Neurocomputing.

[8] Click-Through Rate (CTR), https://www.kaggle.com/datasets/louischen7/2020-digix-advertisement-ctr-prediction, 2022.