Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd

ISSN,e-ISSN: 2548-1304
Volume:8, Issue:2, pp:57-65, 2023
https://doi.org/10.53070/bbd.1223978
Research Paper

Sunucuların Anomali Durumlarinin Yapay Zeka Metotları ile Tahmin Edilmesi

Prediction of Anomaly Conditions of Servers with Artificial Intelligence Methods

Mehmet Fatih SAVRAN*, Ahmet A. MÜNGEN

Yazılım Mühendisliği, OSTİM Teknik Üniversitesi, Ankara, Türkiye

(200801004@ostimteknik.edu.tr, ahmetanil.mungen@ostimteknik.edu.tr)

Received:Dec.24,2022 Accepted: Feb.9,2023 Published:Dec.20, 2023

Özetçe— Sunucularda anormallik tespiti belirli bir metot ve uygulama yoluyla aykırı değerlerin diğerlerinden ayrıştırılarak analiz edilmesiyle oluşan ve genelde olağan dışı durumları tespit etmekte kullanılır. Anormalliklerin erkenden tespit edilmesi ön görülebilirlik kararlar vermeyi ve gerekirse savunma mekanizması geliştirilmesinde kullanılabilir. Önemli bir problem olarak bilinen Anormallik Tespiti birçok tarama ve uygulama sahasında araştırılmaktadır. Genelde araştırmacılar bu bahsi geçen probleme yapay zeka, makine öğrenimi ve durum makine modellemesi gibi teknikleri kullanarak çözüm arayışına girmişlerdir. Sunucuların anormallik testleri ve analizi yapılabilir ve bu yöntem-teknikler kullanılarak çıkarımlar yapılabilir. Sunucudan alınan CPU, Network, Disk, Memory değerleri anomali testinde kullanılmak üzere veri analiz aşamalarından geçer ve teknikler uygulanarak modellemesi yapılır. Bu çalışmada toplanan veri kümesi kullanılarak YSA, Karar Ağacı, Rastgele Orman, K- En Yakın Komşu ve Ekstra Karar Ağacı algoritmalarının anormali tespit performansları test edilmiştir. Yapılan testlerde anormal durumlarının belirlenmesinde % 99.94 oranıyla YSA'nın başarılı olduğu görülmüştür. Önerilen yöntem, toplanan veri ve önerilen yöntemin diğer yöntemler ile karşılaştırmalı analizleri çalışma içerisinde sunulmuştur.

Anahtar Kelimeler : *Anormallik Tespiti, Yapay Zeka, Sunucular, Bulut Sistemler*

Abstract— The objective of this paper is to investigate the efficacy of various techniques in detecting anomalies in server systems. Anomaly detection is a critical problem that involves identifying unusual situations by analyzing outliers and making predictions based on these observations. The study focuses on using artificial intelligence, machine learning, and state machine modeling to solve this problem. The data collected from the servers, including CPU, network, disk, and memory values, is analyzed and used for anomaly testing and modeling. The performance of five algorithms, including Artificial Neural Network (ANN), Decision Tree, Random Forest, K-Nearest Neighbor, and Extra Decision Tree, is evaluated using the collected data set. The results show that the ANN algorithm achieved a success rate of 99.94% in detecting abnormal conditions. The study presents a comprehensive analysis of the proposed method, the collected data, and a comparison of the proposed method with other methods. The findings of this study contribute to the ongoing efforts to improve the accuracy and efficiency of anomaly detection in server systems.

Keywords : *Anomaly Detection, Artificial Intelligence, Servers, Cloud System*

1. Giriş

Anomali Tespiti, bir veri kümesinde olağanın dışındaki durumların ya da kalıpların tespit edilmesini sağlayan bir yöntemdir. Bu olağan dışı durumlar veya kalıplar, bir verinin olağan davranışlarına aykırı durumlar ve kalıplardır. Bu olağan dışı beklenmedik durumlara outliers (aykırı değerler), exceptions (istisnai durumlar) veya anomaliler denir. Başka bir bakış açısıyla, gözlenen bir veya daha fazla değerin önceki çalışma koşullarında olağan kabul edilen davranışların dışında ortaya çıkması olarak ifade edilebilir. Anomaliler sınıflandırma olarak olumlu ya da olumsuz olarak tanımlanamamaktadır. Yalnız belirli bir zaman aralığında, beklenen değerde oluşan dengesizliklerdir. Birçok veri seti sürekli olarak web günlüklerinden, finansal işlemlerden, sağlık kayıtlarından ve

Bu yayın yazarlardan Mehmet Fatih SAVRAN'ın yüksek lisans tez çalışmasından türetilmiştir.

gözetim günlüklerinden ve ayrıca iş, telekomünikasyon ve biyolojik bilimlerden gelir(Aggarwal, 2013) (Jiang et al., 2016). Verilerin çokça ve dağınık özelliğini tanımlayan bir olgu olan "büyük veri" olarak isimlendirilen bu alan, yakın zamanda ve halen bilimin odaklandığı alan haline gelmiştir. Yakın zamanda, büyük verinin başlıca zorlukları büyük ölçekte belirlenmiştir. Bu büyük bir veri değeri, doğruluğu, çeşitli hız ve hacim. Değer, verilerin analizi sonucunda birbirleriyle bağlantılı faydaları ifade etmektedir; doğruluk, verilerin doğruluğunu ifade etmektedir; ve çeşitlilik, yapılandırılmış, yarı yapılandırılmış veya yapılandırılmamış gibi birçok veri türünü ifade etmektedir. Burada hacim elde edilen ve biriken verinin miktarını belli etmektedir. Biriken verinin boyutu ne kadar büyük olursa, hacim de o derecede büyük olmaktadır. Boyutluluk, yapılacak analiz için bulunan verilerdeki özelliklerin ya da değerlerin ya da değişkenlerin sayısını belirtir. Buna karşın hız, verilerin üretildiği "hız" anlamındadır. Hız birçok boyutta bulunabilir. Bulunan büyük veri tanımının bu olguları, zorlukları ele almaktadır. Bu şekilde bir tanım daha farklı bir önemli tarafı göstermemektedir: gerçek dünya veri analizinde başlıca etkenlerden olan "boyutluluk" (Thudumu et al., 2020; Zhai et al., 2014). Boyutlardaki ya da özelliklerdeki ya da niteliklerdeki fazlalaşma, büyük veri kümelerinde anormallik tespiti için engeller çıkarır.

Her durumda, bir anormalliği tanımlayan şey, numuneye ve ölçüm yöntemine bağlıdır. Genel olarak, üç farklı anomali türü ayırt edilir: nokta anomalileri, toplu anomaliler ve bağlamsal anomaliler. Nokta anormallikleri, veri kümesinin geri kalanıyla karşılaştırıldığında anormal görünen bireysel veri örnekleridir. İlişkili veri örnekleri koleksiyonu, veri kümesinin geri kalanına göre anormal ise, toplu anomali olarak adlandırılır.

2. Literatür Taraması

Önceki yapılan araştırmalar, veri madenciliği ve makine öğrenimi yaklaşımlarına dayalı çevrimiçi algılamayı desteklemek için gerçek zamanlı veri toplama için ölçeklenebilir yöntemler oluşturulmuştur (Aguilera et al., 2003; Bahl et al., 2007; Q. Guan et al., 2013; Kiciman & Fox, 2005; Massie et al., 2004; Sigelman et al., 2010; Wang et al., 2011)

Wang et al. (Yalagandula & Dahlin, 2004) çevrimiçi analize izin vermek için EbAT sistemini önerdi sistem düzeyindeki bileşenlerden (örneğin, CPU kullanımı, bellek kullanımı, işletim sisteminin okuma/yazma sayıları, vb.) elde edilen çoklu metriklerin toplamı. Sistem algılama doğruluğunda ve ölçeklenebilirliği izlemede potansiyel gösterdi, ancak yeterince pragmatik bulut senaryoları bağlamında değerlendirilmemiştir. Guan et al. (Y. Guan & Bao, 2009) ve Garfinkel ve al.(Garfinkel & Rosenblum, 2003) önerilen çok seviyeli anomali tespiti bir bulut sisteminin farklı seviyelerindeki izinsiz girişleri tespit etme teknikleri. Bu teknikler oldukça esnek görünmemektedir ve bu tekniklerin operasyonel bağlamda uygulanması daha iyi açıklama gerektirmektedir. Lee et al. (Lee et al., 2011) anomalilerin hızlı tespitini sağlayan çok seviyeli bir yaklaşım önerdi her konuk işletim sisteminin sistem günlüklerinde bulunur. Dezavantajlarından biri de bariz ölçeklenebilirlik eksikliği, çünkü giderek daha fazla kaynak gerektiriyor yüksek sistem iş yükü altında. Ayrıca, yalnızca günlük veri kaydında algılamaya özeldir. Benzer şekilde, Dastjerdi et al. (Dastjerdi et al., 2009) mobil tabanlı bir yaklaşım önerdi bulut sistemleri için bir saldırı tespit sistemi için aracılar. Ancak, aracıya bağlanması gereken çok sayıda sanal makine nedeniyle ölçekleme yeteneği bir sorun olarak görünmektedir. E2EPrf ve SysProf, sırasıyla (Agarwala et al., 2007; Agarwala & Schwan, 2006) önerilen ve farklı ayrıntı düzeylerinde izleme bilgilerini yakalayabilen profil oluşturma araçlarıdır.

PREPARE (Tan et al., 2012) ve DAPA (Kang et al., 2012), yakın zamanda sanallaştırılmış ortamlar için anormallik algılamaya dayalı performans değerlendirmesi için önerilen iki çerçevedir. Ancak, bu yaklaşımların hiçbiri, sanal makine canlı geçişi ve bulutun heterojenliğinden kaynaklanan yüksek hacimli veriler gibi bulutun esnekliğinin etkisine odaklanmaz. Tablo 1'de incelenen çalışmaların ortak ve farklı özellikleri bir gösterilmiştir.

Ling Huang ve arkadaşları (Huang et al., 2007) yaptıkları "In-Network PCA and Anomaly Detection" adlı çalışmada anormallikleri önceki yöntemlerden çok daha az veriyle tespit etmek için dağıtılmış izlemeyi PCA analiziyle birleştiren ağ anormalliği tespiti için yeni bir algoritmik çerçeve sunmuşlardır. Ağ anormalliği tespiti için, anormallikleri önceki yöntemlere göre çok daha az veriyle tespit etmek için PCA analizi ile dağıtılmış izlemeyi birleştiren yeni bir algoritmik çerçeve sunduk. kriterler. Buradaki fikir, yerel izleme verilerini yalnızca doğru algılamayı sağlayacak kadar izlemektir. Yerel filtreleme, ağ üzerinden iletilen veri miktarını azaltır, ancak aynı zamanda anormallik tespitinin, küresel durumun sınırlı veya kısmi görünümleriyle yapılması gerektiği anlamına gelir.

K-Means kümeleme (MacQueen, 1967) nesneleri özellik değerlerine göre K olarak gruplayan ayrık kümeler olarak bir kümeleme analiz algoritmasıdır. Anormal davranışı tespit etmek için başka bir yaklaşım, küme yapılarına benzer özelliklere sahip veri noktaları atayarak işlev gören kümelemeye dayanmaktadır (UCI Machine Learning Repository, 2015). Aynı kategoride benzer küme özellik değerlerine sahip sınıflandırılan nesnelerdir.

Tablo 1. Araştırılan Sistem Çalışmaların Özeti

Referans	Ölçeklenebilirlik	Çevrimiçi	Belleksiz	Çok Düzeyli
EbAT/Wang et al.				
Guan et al.				
Garfinkel et al.				
Lee et al.				
Dastjerdi et al.				
E2EProf				
SysProf				
PREPARE				
DAPA				

	→	Uygun
	→	Uygun Değil
	→	Değişkenlik Gösterir

K-Means+C4.5 sınıflandırıcısının performansı değerlendirilmiştir ve bunu, altı performans ölçüsü kullanarak bireysel K-Means kümeleme ve C4.5 karar ağacı yöntemleriyle karşılaştırılmıştır. Sınıflandırma performansını iyileştirmek için iki başarılı veri bölme yöntemini basamaklandırmak için yeni bir yöntem sunmaktadır. Anormallik algılama perspektifinden bakıldığında, çalışma yüksek performanslı bir anormallik algılama sistemi sunmayı amaçlanmıştır. Açık kaynaklı bir makine öğrenimi çerçevesi Weka (Weka 3.5) (University of Waikato, 2016) kullanılmıştır. Weka, veri madenciliği uygulamaları için bir makine öğrenme algoritması koleksiyonudur.

K-Means, ID3, Naïve Bayes, K–NN, SVM, TCM–KNN' nin performansını ve çalışmada (Muniyandi et al., 2012) kullanılan 41 özellik için KDD99 veri seti için 5 denemenin ortalaması alınan K–Means ve C4.5 algoritmalarının önerilen kademeli algoritmasını göstermektedir.

3. Önerilen Yöntem

Çalışma kapsamında toplanan veri setimizde toplam 286 bin 440 adet veri bulunmaktadır. Bu veriler belirli bir süre içinde belirli aralıklarla birden fazla sunucudan alınan anlık kayıtlardan oluşmaktadır. Veriler işleme tabi tutulmadan önce veri tabanı üzerinde bazı derlenmeler sonucunda Excel olarak çıktısı alınabilir duruma getirilmiştir. Veri setine ait öznitelikler ve açıklamaları Tablo 2'de görülmektedir.

Tablo 2. Veri Seti Detayları

BağımsızDeğişkenler	Açıklama	Düzeyleri
Id	Benzersiz kimlik	Sürekli
HostId	Verilerin elde edildiği sunucu	Kategorik
AvailablePhyslMem	Kullanılabilir bellek	Sürekli
CpuPercent	İşlemci kullanım yüzdesi	Sürekli
CpuTotalHits	-	Sürekli
MemoryFreePerc	Kullanılabilir bellek yüzdesi	Sürekli
MemoryOccupiedPerc	Kullanılan bellek yüzdesi	Sürekli

MemoryTotal	Toplam bellek	Sürekli
Time	Zaman	Sürekli
DiskName	Verilerin alındığı disk	Kategorik
TotalSizeByt	Toplam saklama alanı	Sürekli
AvailFreeSpaceByt	Boş saklama alanı	Sürekli
UsedSizeByt	Kullanılan saklama alanı	Sürekli
UsagePerc	Saklama alanı kullanım yüzdesi	Sürekli
BytesReceived	Ağda gelen veri	Sürekli
BytesSent	Ağda gönderilen veri	Sürekli
NetworkName	Ağ kartının ismi	Kategorik
OpsStatus	Ağ durumu	Kategorik
SpeedBitsPerSec	Ağ hızı	Sürekli
BağımlıDeğişkenler		
isAnomaly	Anomali durumu	Kategorik

Boş, tanımlanmamış veya değer barındırmayan özellikler ve veriler tespit edilmiştir. Bunun sonucunda bazı özelliklerin ve verilerin temizlenmesi sağlanmıştır. "CpuTotalHits" özelliğinin bulundurduğu verilerin boş olduğu tespit edilmiş ve bu özellik verilerin arasından çıkarılmıştır. Boş veriler ise 10 taneyi geçmemekle birlikte veri seti içerisinden silinmiştir. Yapay zeka algoritmalarının çalışabilmesi için verilerin sayısallaştırılması gereklidir. Burada amaç genellikle muhtelif verileri tanımlayarak, betimleyerek ya da farklı verileri karşılaştırarak ortak çıkarımlar üretmektir (Lewis, 2015). Sayısallaştırma uygulaması yapılarak verilerimiz işlenebilmeye uygun bir hale getirilmiştir. Tablo 3' de bu nedenle bu aşamada gerekli sınıflandırma işlemleri yapıldı.

Tablo 3. Veri Seti Özelliklerin Sayısallaştırılması

	Deger	Sayisal
HostId	2a5824c8-6512-48be-824f-412ee1ffa69d	1
	bfb7919d-c841-4f33-8e79-33ae74dcde22	2
	0a7f60ca-6a2a-4da5-a8b7-e8597c2b7405	3
	d4e7094c-d0e8-4485-9c67-55927d203d03	4
OpsStatus	BMRU	0
	BMsRU	1
	Up	2
	BMU	3
isAnomaly	DOĞRU	1
	YANLIŞ	0

Veri setinde bulunan her öznitelik için özniteliklerin değerlerinin değişim aralığını gösteren grafiklerinin incelenmesi yapıldı. Veri setindeki veriler, her öznitelik için ayrı ayrı grafiklendirilerek incelenmiştir. Bu inceleme sonucunda bazı gürültü niteliğinde verilere ulaşılmıştır. Gürültülü veri, veri kümesi içinde yer alan ama veri madenciliği uygulamasında kullanılmayacak ve bir anlam içermeyen verilerdir (Han et al., 2012) . Bu verilerin grafik incelenmesi yapılmış ve gürültü niteliğindeki veri setini etkilemeyecek veriler her tablo için belirlenerek temizlenmiştir. Elde edilen son veriler üzerinden özelliklerin değerlerinin analizi yapıldı. Her öznitelik için minimum, maksimum, ortalama ve standart sapma değerleri bulundu (Tablo 4).

Veri madenciliğinde, tahmine dayalı bir modelin sonuçlarındaki değişkenliği veya belirsizliği ölçmek için standart sapma kullanılır. Standart sapma olarak da bilinen standart sapma, veri madenciliği uygulamalarında yaygın olarak kullanılan istatistiksel bir ölçüdür. Bir dizi veri noktasının ortalamaları veya ortalamaları etrafındaki yayılmasının veya dağılımının bir ölçüsüdür.

Standart sapma, farklı modellerin değişkenliğini karşılaştırmak veya veri setinin geri kalanından önemli ölçüde farklı olan veri noktalarını belirlemek için de kullanılabilir. Standart sapmanın yorumlanması, özel uygulamaya ve kullanıldığı bağlama bağlıdır. Bununla birlikte, genel olarak, bir veri madenciliği uygulamasının sonuçlarının belirsizliğini ve değişkenliğini anlamak için yararlı bir araçtır. Bu çalışmada standart sapma, varyansın karekökü olarak hesaplanmıştır. Varyans, her bir veri noktasının ortalamadan ne kadar uzakta olduğunun bir ölçüsüdür ve ortalamadan farkların karelerinin ortalaması olarak hesaplanır.

Tablo 4. Veri Seti Özellik Değer Analizi

Özellik Adı	Minimum	Maksimum	Ortalama	Standart Sapma
AvailablePhyslMem	462948	15338868	13301684	2458695,131
CpuPercent	0	99,24	49,62	35,25
MemoryFreePerc	0	92,71	46,35	36,21
MemoryOccupiedPerc	0	98,97	49,48	32,73
MemoryTotal	939224	16776192	8857708	8968586,412
TotalSizeGB	0,48	126,51	63,49	51,58
AvailFreeSpaceGB	0,26	97,02	48,64	41,34
UsedSizeGB	0,008	45,89	22,95	19,45
UsagePerc	45	96	48	33,68
GBReceived	0	190,17	95,08	12,93
GBSent	0	138,44	69,22	17,59
OpsStatus	0	3	1,5	1,29
SpeedGBPerSec	0	4,65	2,32	2,22
isAnomaly	0	1	0,5	0,7

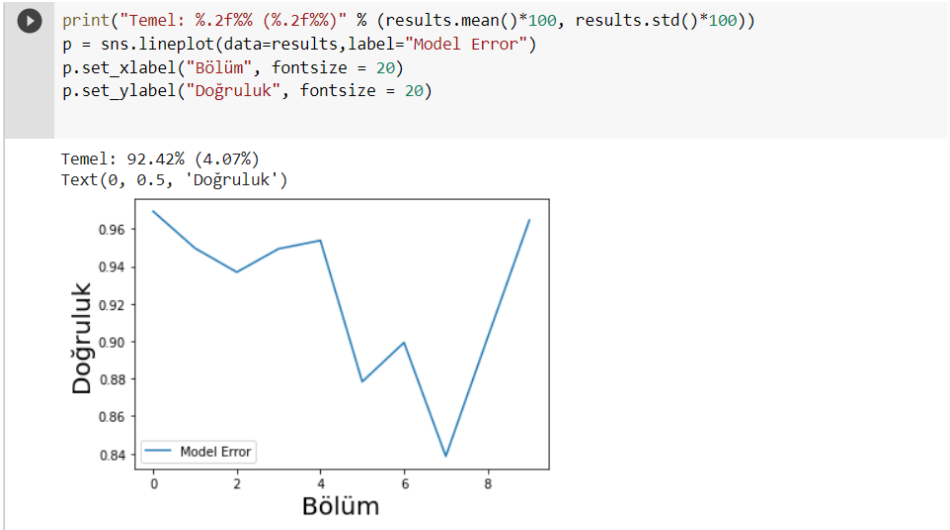
3.2. Modelleme

Keras, Python ile yazılmış, makine öğrenimi platformu TensorFlow üzerinde çalışan bir derin öğrenme API'sidir . Hızlı denemeyi mümkün kılmaya odaklanarak geliştirilmiştir. Bir fikirden sonuca olabildiğince hızlı gidebilmek, iyi araştırma yapmanın anahtarıdır. Keras kaynağı düşünülen bir problem üzerindeki geliştiricinin bilişsel yükünü azaltmak için kurgulanmıştır. Keras, karmaşıklığın kademeli olarak çözüme ulaştırmasını sağlamak için geliştirilmiştir. Hızlılık ve kolaylığı bünyesinde kullanılabilir olarak tasarlamıştır. Keras, TensorFlow 2'nin üst düzey API'sidir: Modern derin öğrenmeye odaklanan, makine öğrenimi sorunlarını çözmek için ulaşılabilir, son derece üretken bir arayüz. Yüksek yineleme hızıyla makine öğrenimi çözümleri geliştirmek ve göndermek için temel soyutlamalar ve yapı taşları sağlar (Team, n.d.). Şekil 1'de uygulamada kullanılan Python kodlarından biri gösterilmiştir.

```
+ Kod  + Metin

!pip install scikeras[tensorflow]
!pip install tensorflow==2.7.0

from pandas import read_csv
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from scikeras.wrappers import KerasClassifier
from sklearn.model_selection import cross_val_score
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import StratifiedKFold
# load dataset
dataframe = read_csv("FinalDataFrom.csv", header=None)
dataframe.drop_duplicates(keep = False)
dataset = dataframe.values
# split into input (X) and output (Y) variables
X = dataset[:,0:17].astype(float)
Y = dataset[:,17]

# baseline model
def create_baseline():
  # create model
  model = Sequential()
  model.add(Dense(17, input_shape=(17,), activation='relu'))
  model.add(Dense(1, activation='sigmoid'))
  # Compile model
  model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
  return model
# evaluate model with standardized dataset
estimator = KerasClassifier(model=create_baseline, epochs=10, batch_size=64, verbose=0)
kfold = StratifiedKFold(n_splits=10, shuffle=True)
results = cross_val_score(estimator, X, Y, cv=kfold)
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

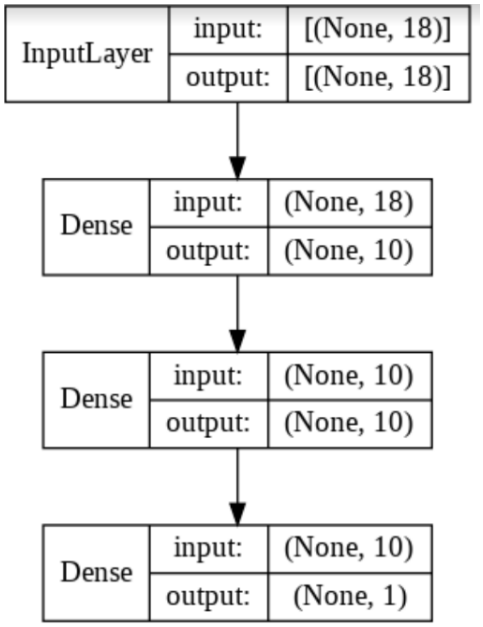
Şekil 1. Modelleme için Başvurulan Kodlama

Keras ile derin öğrenme modellemesi yapmak, diğer yöntem ve platformlara göre daha az karmaşık ve kullanışlıdır. Belirli bir sıralama ile başarılı bir modelleme işlemi tamamlanabilir. İlk olarak eğitim verisi tanımı yapılmalıdır. İkinci olarak eğitim verisinin tanımlanmasının ardından modelin ve katmanlarının tanımının yapılması gerekir. Daha sonra Epoch yani Tur sayısı, Loss Fonksiyonu ve optimizer parametrelerini girilmelidir. Son olarak ise model, eğitim verileriyle beslenmeli ve öğrenme işlemi sürdürülmelidir. Elde edilen sonuçlar isimlendirme yapılarak Şekil 2' deki gibi çıktı elde edilmiştir. X eksenine yerleşen değerler Doğruluk değişimini, Y eksenine yerleşene değerler ise Bölüm adedinin değişimini göstermektedir.



Şekil 2. Modelleme Çıktı Ekranı

Keras kütüphanesi uygulanan modellemenin katmanlarını göstermek için de yetkindir. Katmanları şekillerle birlikte ekranda görebilmek için kütüphane tanıtılmalı, model oluşturulmalı ve "plot_model" fonksiyonu ile gerekli parametreler girilerek çıktı olarak katmanlar elde edilir.



Şekil 3. Uygulanan Modelleme Katman Görünümü

Makine Öğrenmesinde modelleme işlemi, gösterilen tanıtılmış bir görevi elde edilmiş veriler ile kendini eğiterek uygulayan bir algoritma olarak tanımlanır. Bir sirkülasyon içerisinde ileri ver geri yayılımlar yaparak modelin ağırlıklarını ve biasını edinme, kendini eğitme amacı ile değişim sağlar. Aktivasyon fonksiyonu, bir karar mekanizması olarak adlandırılır. Bu karar mekanizması ağırlıklı toplamı hesaplar ve gerekirse bu toplama daha fazla bias ekleyerek nöronun aktif edilip edilmemesine karar vermektedir. Genel olarak aktivasyon fonksiyonunun amacı, nöronun verdiği çıktıyı doğrusal olmaktan kurtarmaktır. Bu sayede aktivasyon fonksiyonlarının, bu çalışmada alınan çıktının doğruluğuna ve başarımına katkısı çoktur. Modellemede bulunan bir diğer etken ise Loss Fonksiyonudur. Bu fonksiyon belirli bir algoritmanın girilen verileri ne kadar iyi modellediğini hesaplamaya yaramaktadır.

Yapay Sinir Ağları'nın yapısında 3 adet katman bulunmaktadır. Bunlar; Giriş, Gizli ve Çıkış katmanlarıdır. Gizli(Hidden) Katman, Giriş ve Çıkış katmanı arasında bulunmaktadır. Katmanlarda belirli sayılarda düğüm(node) bulunmaktadır. Bu düğümler sahip oldukları ağırlıkla birlikte modelin tamamını oluşturmaktadır. Diğer modelleme ortamlarında olduğu gibi Python ve Keras' ta bize modelleme çıktılarıyla birlikte katman mimarisinin de çıktısını verebilmektedir. Şekil 3'de görüldüğü gibi oluşturulan modelimizde bulunan 3 katmanlı mimarinin görselleştirilmiş ve tanımlanmış hali bulunmaktadır. Çalışmamızda veriler test ve eğitim olarak gruplandırılmış ve bunlar modellemeye sunulmuştur. Bu veriler ile ileri geri yayılım sayesinde model eğitilmiş ve belirli bir başarım oranı elde edilmiştir. Bu modellemenin çıktıları girilen parametrelerine göre değişim gösterebilmektedir. Bu nedenle birden çok denemelerin içinden başarım oranının en yüksek olanı bu çalışmada sunulmuştur. Yüzde 92,42 olarak elde edilen başarım oranı ile literatürdeki çalışmaların tavan başarım oranlarına yakın ve bir çoğundan daha yüksek oran elde edilmiş olundu.

Girilen çeşitli parametreler ile farklı yöntemlerle elde edilen çıktılar Tablo 5'de gösterilmiştir. Farklı metotlar kullanılarak elde edilen modellemelerin her bir başarım oranı listelenmiştir. Buradan çıkarımla başarım oranları ve modellemenin avantaj ve dezavantajları da bilindiği varsayılarak bir sonuç ya da öngörüye varılabilir. Toplamda 5 adet farklı metot denenmiş olup aralarındaki farklılıklar net bir şekilde ortaya konabilmiştir. Daha önceki çalışmaların çıktılarıyla da kıyaslanarak çalışmamızın değeri anlaşılabilir.

Çapraz doğrulama ortalama başarı oranı, tahmine dayalı bir modelin doğruluğunu değerlendirmek için veri madenciliği uygulamalarında kullanılan bir performans ölçüsüdür. Çapraz doğrulama, mevcut verileri birden çok alt kümeye bölmeyi, modeli bir alt kümede eğitmeyi ve kalan alt kümelerde doğruluğunu test etmeyi içeren bir yeniden örnekleme tekniğidir. Ortalama başarı oranı, çapraz doğrulama sürecinin her yinelemesinden elde edilen doğruluk puanlarının ortalaması alınarak hesaplanır. Çapraz doğrulama ortalama başarı oranı, yeni, görünmeyen veriler üzerinde modelin beklenen doğruluğunun bir tahminini sağlar. Bir model eğitim verilerine çok yakın olduğunda ve yeni veriler üzerinde zayıf genelleme performansına sahip olduğunda ortaya çıkan aşırı uydurmanın önlenmesine yardımcı olur.

Makine Öğrenmesinde modelleme işlemi, gösterilen tanıtılmış bir görevi elde edilmiş veriler ile kendini eğiterek uygulayan bir algoritma olarak tanımlanır. Bir dolaşım içerisinde ileri ver geri yayılımlar yaparak modelin ağırlıklarını ve biasını edinme, kendini eğitme amacı ile değişim sağlamaktadır. YSA'lar, girdi değişkenleri ile hedef değişken arasındaki doğrusal olmayan ilişkileri öğrenebilir, bu da onları karmaşık ve doğrusal olmayan

anormallikleri tespit etmek için çok uygun hale getirir. Ayrıca, YSA'lar çok çeşitli kalıpları öğrenebilir ve basit eşiklemeye dayalı karar kurallarıyla sınırlı değildir. Bu, diğer algoritmalar tarafından yakalanmayabilecek değişkenler arasındaki karmaşık ilişkileri yakalamalarına olanak tanır. Buna ek olarak YSA'lar, amacın olağandışı veya görünmeyen durumları belirlemek olduğu anormallik tespit görevlerinde önemli olan yeni, görünmeyen verileri genellemek için tasarlanmıştır. Bu nedenlerle YSA'nın diğerlerine göre daha başarılı olduğu görülmüştür.

Tablo 5. Kullanılan Tekniklere Göre Model Çıktıları

Modellemede Kullanılan Teknik	Çapraz doğrulama ortalama başarım oranı (%)
YSA	**99.94%**
Karar Ağacı	99.44%
Rastgele Orman	99.40%
K- En Yakın Komşu	99.41%
Ekstra Karar Ağacı	99.63%

4. Sonuç

Bu çalışmada, YSA, Karar Ağacı, Rastgele Orman, K-En Yakın Komşu ve Ekstra Karar Ağacı algoritmalarının modellemesi uygulanmış ve bunun sonunda birçok çıktı elde edilmiştir. Modelleme ile birlikte başta YSA olmak üzere diğer kullanılan sınıflandırma algoritmaları çeşitli sonuçlar vermişlerdir. Karar Ağacı ve onun farklı versiyonları olarak tanımlanabilecek; Rastgele Orman ve Ekstra Karar Ağacı algoritmalarının kullanılabilirliği ve çözüm odaklılığı bakımından oldukça verimli olduğu anlaşılmıştır. En Yakın Komşu Algoritması ise basit uygulanabilirliği açısından diğer algoritmalara göre daha el verişli olduğu görülmüştür. Elde edilen birçok sonuçtan başarımı en yüksek yapı başarılı model olarak seçilerek bu çalışmada sunulmuştur. Bu çalışmada sunulanlar ışığında çok sunuculu bir yönetime sahip olan çalışma alanlarında YSA ile oluşturulan model ile bir uygulama tasarlanabilir. Bu uygulama ile derin öğrenme sayesinde sunuculardaki anormal durumların önceden sezinlenmesi ve tespit ile ilgili bir reaksiyon sağlayan uyarı sistemine sahip olunabilir.

Kaynaklar

Agarwala, S., Alegre, F., Schwan, K., & Mehalingham, J. (2007). E2EProf: Automated end-to-end performance management for enterprise systems. *Proceedings of the International Conference on Dependable Systems and Networks*. https://doi.org/10.1109/DSN.2007.38

Agarwala, S., & Schwan, K. (2006). SysProf: Online distributed behavior diagnosis through fine-grain system monitoring. *Proceedings - International Conference on Distributed Computing Systems, 2006*. https://doi.org/10.1109/ICDCS.2006.81

Aggarwal, C. C. (2013). Mining sensor data streams. In *Managing and Mining Sensor Data* (Vol. 9781461463092). https://doi.org/10.1007/978-1-4614-6309-2_6

Aguilera, M. K., Mogul, J. C., Wiener, J. L., Reynolds, P., & Muthitacharoen, A. (2003). Performance debugging for distributed systems of black boxes. *Operating Systems Review (ACM)*, *37*(5). https://doi.org/10.1145/1165389.945454

Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D. A., & Zhang, M. (2007). Towards highly reliable enterprise network services via inference of multi-level dependencies. *Computer Communication Review*, *37*(4). https://doi.org/10.1145/1282427.1282383

Dastjerdi, A. V., Bakar, K. A., & Hassan Tabatabaei, S. G. (2009). Distributed intrusion detection in clouds using mobile agents. *3rd International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP 2009*. https://doi.org/10.1109/ADVCOMP.2009.34

Garfinkel, T., & Rosenblum, M. (2003). A Virtual Machine Introspection Based Architecture for Intrusion Detection. *Proc. Network and Distributed Systems Security …*, *1*.

Guan, Q., Fu, S., de Bardeleben, N., & Blanchard, S. (2013). Exploring time and frequency domains for accurate and automated anomaly detection in cloud computing systems. *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing, PRDC*. https://doi.org/10.1109/PRDC.2013.40

Guan, Y., & Bao, J. (2009). A CP Intrusion Detection Strategy on Cloud Computing. *2009 International Symposium on Web Information Systems and Applications, Proceedings*, *8*.

Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques. In *Data Mining: Concepts and Techniques*. https://doi.org/10.1016/C2009-0-61819-5

Huang, L., Nguyen, X. L., Garofalakis, M., Jordan, M. I., Joseph, A., & Taft, N. (2007). In-network PCA and anomaly detection. *Advances in Neural Information Processing Systems*. https://doi.org/10.7551/mitpress/7503.003.0082

Jiang, F., Leung, C. K., & Pazdor, A. G. M. (2016). Big data mining of social networks for friend recommendation. *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016*. https://doi.org/10.1109/ASONAM.2016.7752349

Kang, H., Zhu, X., & Wong, J. L. (2012). DAPA: Diagnosing application performance anomalies for virtualized infrastructures. *2nd USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services, Hot-ICE 2012*.

Kiciman, E., & Fox, A. (2005). Detecting application-level failures in component-based Internet services. *IEEE Transactions on Neural Networks*, *16*(5). https://doi.org/10.1109/TNN.2005.853411

Lee, J. H., Park, M. W., Eom, J. H., & Chung, T. M. (2011). Multi-level intrusion detection system and log management in cloud computing. *International Conference on Advanced Communication Technology, ICACT*.

Lewis, S. (2015). Qualitative Inquiry and Research Design: Choosing Among Five Approaches. In *Health Promotion Practice* (Vol. 16, Issue 4). https://doi.org/10.1177/1524839915580941

MacQueen, J. B. (1967). Kmeans Some Methods for classification and Analysis of Multivariate Observations. *5th Berkeley Symposium on Mathematical Statistics and Probability 1967*, *1*(233), 281–297. https://doi.org/citeulike-article-id:6083430

Massie, M. L., Chun, B. N., & Culler, D. E. (2004). The ganglia distributed monitoring system: Design, implementation, and experience. *Parallel Computing*, *30*(7). https://doi.org/10.1016/j.parco.2004.04.001

Muniyandi, A. P., Rajeswari, R., & Rajaram, R. (2012). Network anomaly detection by cascading k-Means clustering and C4.5 decision tree algorithm. *Procedia Engineering*, *30*. https://doi.org/10.1016/j.proeng.2012.01.849

Sigelman, B. H., Andr, L., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., Jaspan, S., & Shanbhag, C. (2010). Dapper , a Large-Scale Distributed Systems Tracing Infrastructure. *Google Research*, *April*.

Tan, Y., Nguyen, H., Shen, Z., Gu, X., Venkatramani, C., & Rajan, D. (2012). PREPARE: Predictive performance anomaly prevention for virtualized cloud systems. *Proceedings - International Conference on Distributed Computing Systems*. https://doi.org/10.1109/ICDCS.2012.65

Team, K. (n.d.). *Keras Developer Guides*. Retrieved August 24, 2022, from https://keras.io/guides/

Thudumu, S., Branch, P., Jin, J., & Singh, J. (Jack). (2020). Adaptive Clustering for Outlier Identification in High-Dimensional Data. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11945 LNCS*. https://doi.org/10.1007/978-3-030-38961-1_19

UCI Machine Learning Repository. (2015). KDD Cup 1999 Data. In *1999]*. *Http://Kdd. Ics. Uci. Edu/Databases/Kddcup99/Kddcup99. Html*.

University of Waikato. (2016). Weka 3 - Data Mining with Open Source Machine Learning Software in Java. In *The University of Waikato*.

Wang, C., Viswanathan, K., Choudur, L., Talwar, V., Satterfield, W., & Schwan, K. (2011). Statistical techniques for online anomaly detection in data centers. *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, IM 2011*. https://doi.org/10.1109/INM.2011.5990537

Yalagandula, P., & Dahlin, M. (2004). A Scalable Distributed Information Management System ∗ Categories and Subject Descriptors. *Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*.

Zhai, Y., Ong, Y. S., & Tsang, I. W. (2014). The emerging ?Big dimensionality? *IEEE Computational Intelligence Magazine*, *9*(3). https://doi.org/10.1109/MCI.2014.2326099