



Teknoloji Fakültesi

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Derin öğrenme ve Makine öğrenme ile veri tabanlarında aykırılıklar tespit etme ve erken müdahale sistemi

Aisan KHEIRI

Kaan Kürşat KILIÇ

"

Bitirme Projesi 1. Ara Raporu

Bilgisayar Mühendisliği Bölümü

DANIŞMAN

Doç. Dr. Buket DOĞAN

İSTANBUL, 2025

"

İÇİNDEKİLER

1. GİRİŞ	8
1.1. Proje Çalışmasının Amacı ve Önemi	8
▪ 1.1.1 Çalışmanın Amacı	8
▪ 1.1.2 Çalışmanın Önemi	9
▪ 1.1.3 İlgili Çalışmalar	10
1.2. Benchmark Çalışmaları	14
▪ 1.2.1. Patent Araştırması	14
▪ 1.2.2. Patent Araştırma Sonuçları	15
▪ 1.2.3. Sonuç ve Değerlendirme	15
2. MATERYAL VE YÖNTEM	16
2.1. Veri Seti ve Özellikleri	16
▪ 2.1.1. Veri Setinin İçeriği	16
▪ 2.1.2. Veri Ön İşleme ve Temizleme	17
▪ 2.1.3. Veri Setinin Kullanım Amacı	17
▪ 2.1.4 Veri Seti Örneği	17
2.2. Makine Öğrenmesi ve Derin Öğrenme Algoritmaları	17
▪ 2.2.1 Makine Öğrenmesi Algoritmaları ve Hiperparametreler	18
▪ 2.2.2 Kullanılan Makine Öğrenmesi Sonuçları	20
▪ 2.2.3 Kullanılması Planlanan Derin Öğrenme Algoritmaları	21
3. WEB SİTESİ GELİŞTİRME VE KULLANILAN TEKNOLOJİLER	22
3.1. Web Sitesinin Genel Yapısı	22
3.2. Kullanılan Teknolojiler ve Araçlar	23
3.3. Web Sitesine Ait Görseller	25

4. BULGULAR VE TARTIŞMA	26
4.1. Şu Ana Kadar Yapılan Çalışmalar	26
▪ 4.1.1 Veri Toplama ve Ön İşleme	26
▪ 4.1.2 Web Sitesinin Geliştirilmesi	26
▪ 4.1.3 Kullanılan Makine Öğrenmesi Modelleri	26
▪ 4.1.4 Model Performans Değerlendirmesi	27
4.2. Gelecekte Yapılacak Çalışmalar	27
▪ 4.2.1 Kullanılması Planlanan Modeller	27
▪ 4.2.2 Web Sitesine Yeni Özellikler Eklenmesi	27
5. SONUÇ	27

ÖZET

Bu çalışma, veri tabanı sistemlerinde ortaya çıkabilecek olası sorunları önceden tespit ederek, otomatik müdahale mekanizmaları geliştirmeyi hedeflemektedir. Bu doğrultuda, makine öğrenimi ve derin öğrenme tabanlı anomali tespit yöntemleri kullanılmaktadır. Araştırma kapsamında sürekli toplanan telemetri verileri detaylı bir şekilde analiz edilerek ön işleme aşamalarından geçirilmiş ve değerlendirilmektedir. Veri tabanlarında oluşabilecek hataları erken tespit edebilmek için yerli bir yazılım çözümü geliştirilmesi hedeflenmiştir.

Ayrıca, bu sistemin bir web sitesi üzerinden erişilebilir olması planlanarak kullanıcıların model eğitim sürecine dahil edilmesi amaçlanmaktadır. Kullanıcılar, hiperparametreleri belirleyerek model eğitimi gerçekleştirebilecek ve sonuçları değerlendirebilecektir. Çalışma kapsamında veri toplama, ön işleme, analiz ve model değerlendirme süreçleri ayrıntılı bir şekilde ele alınmış, farklı makine öğrenimi ve derin öğrenme algoritmalarının performans karşılaştırmaları yapılmıştır.

Sonuç olarak, bu çalışma, veri tabanı yönetim süreçlerini optimize ederek verimliliği artırmayı ve gelecekteki otomasyon sistemlerine katkı sağlamayı amaçlamaktadır. Projenin uzun vadeli hedefleri arasında, veri tabanlarını daha verimli hale getirerek sorgu maliyetlerini en aza indirmek ve sistemin, sunucu içindeki gerekli tüm aksiyonları otomatik olarak gerçekleştirebilir duruma gelmesini sağlamak yer almaktadır. Ayrıca, elde edilen analiz sonuçlarının raporlanarak sistematik bir şekilde saklanması planlanmaktadır.

Anahtar Kelimeler: Veri Tabanı Yönetimi, Anomali Tespiti, Telemetri Analizi, Veri Madenciliği, Büyük Veri, Makine Öğrenmesi Modelleri, Derin Öğrenme Algoritmaları, Hiperparametre Optimizasyonu, Yapay Zeka Tabanlı Analiz, Veri Ön İşleme Teknikleri, Model Performans Ölçümü, Otomatik Anomali Algılama, Web Tabanlı Veri Analizi, Sunucu Sağlık Kontrolü, Akıllı Veri Yönetimi.

Mart, 2025

Aisan KHEIRI, Kaan Kürşat KILIÇ

ABSTRACT

This study aims to detect potential issues in database systems in advance and develop automated intervention mechanisms. In this context, anomaly detection methods based on machine learning and deep learning are utilized. As part of the research, continuously collected telemetry data is thoroughly analyzed, preprocessed, and evaluated. A domestic software solution is intended to be developed to detect possible database errors at an early stage.

Additionally, making this system accessible via a website is planned, allowing users to participate in the model training process. Users will be able to define hyperparameters, train models, and evaluate the results. The study extensively covers data collection, preprocessing, analysis, and model evaluation processes while comparing the performance of various machine learning and deep learning algorithms.

As a result, this study aims to optimize database management processes, enhance efficiency, and contribute to future automation systems. The long-term objectives of the project include improving database efficiency, minimizing query costs, and enabling the system to autonomously execute all necessary actions within the server. Furthermore, it is planned that the obtained analysis results will be systematically recorded and reported.

Keywords: Database Management, Anomaly Detection, Telemetry Analysis, Data Mining, Big Data, Machine Learning Models, Deep Learning Algorithms, Hyperparameter Optimization, AI-Based Analysis, Data Preprocessing Techniques, Model Performance Evaluation, Automated Anomaly Detection, Web-Based Data Analysis, Server Health Monitoring, Intelligent Data Management.

March, 2025

Aisan KHEIRI, Kaan Kürşat KILIÇ

KISALTMALAR

DBMS	: Database Management System
DBA	: Database Administrator
LLM	: Large Language Model
ML	: Machine Learning
DL	: Deep Learning
AI	: Artificial Intelligence
SQL	: Structured Query Language
RDBMS	: Relational Database Management System
API	: Application Programming Interface
GPU	: Graphics Processing Unit
TPU	: Tensor Processing Unit
ORM	: Object-Relational Mapping
FPDF	: Free PDF Library for Python
RNN	: Recurrent Neural Network
LSTM	: Long Short-Term Memory
SVM	: Support Vector Machine
KNN	: K-Nearest Neighbors
AutoML	: Automated Machine Learning

ŞEKİL LİSTESİ

Şekil 2.1: Veri Setinin İlk Beş Satırı	17
Şekil 3.1: Yüklenmesi Gereken Veri Seti Dosyası	25
Şekil 3.2 : Algoritma Seçimi	25
Şekil 3.3 : Veri Ön İşleme Aşamasında Hiperparametre Seçimi	26

TABLO LİSTESİ

Tablo 1.1 Literatür Tarama	10
Tablo 2.1 Üç Algoritmaya Ait Model Eğitim Sonuçları	20

1. GİRİŞ

Günümüzde veritabanı yönetim sistemleri, büyük ölçekli şirketlerden kamu kuruluşlarına kadar birçok alanda kritik bir rol oynamaktadır. Veritabanlarının kesintisiz ve hatasız çalışması, veri güvenliği, sistem performansı ve iş sürekliliği açısından büyük önem taşımaktadır. Ancak, sistemlerde meydana gelebilecek anormal durumlar (aykırılıklar), performans düşüşlerine, veri kayıplarına veya güvenlik tehditlerine yol açabilmektedir. Geleneksel yöntemler genellikle statik kural tabanlı tespit sistemlerine dayanmakta olup, yeni ve bilinmeyen anomalileri algılamakta yetersiz kalmaktadır. Bu nedenle, makine öğrenmesi (ML) ve derin öğrenme (DL) tabanlı otomatik tespit ve müdahale mekanizmalarına duyulan ihtiyaç giderek artmaktadır.

Bu çalışmada, makine öğrenmesi ve derin öğrenme teknikleri kullanılarak, veritabanlarında meydana gelen anormalliklerin tespit edilmesi ve otomatik müdahale mekanizmalarının geliştirilmesi amaçlanmaktadır. Sistem, veritabanı telemetri verilerini sürekli olarak analiz ederek, anomalileri erken aşamada tespit edecek ve proaktif önlemler alacaktır. Makine öğrenmesi algoritmaları, geçmiş verilere dayalı istatistiksel analiz yaparak anomali modellerini belirlerken; derin öğrenme teknikleri, daha karmaşık ve öngörülemeyen anomalileri tespit etmek için gelişmiş yapay sinir ağları ile çalışacaktır.

1.1. Proje Çalışmasının Amacı ve Önemi

▪ 1.1.1 Çalışmanın Amacı

Bu çalışma, makine öğrenmesi (ML) ve derin öğrenme (DL) tekniklerini kullanarak veritabanlarında meydana gelen anomalileri tespit etmek ve erken müdahale mekanizmaları geliştirmek amacıyla yürütülmektedir. Veritabanı sistemleri, günümüz dijital dünyasında büyük ölçekli işletmelerden kamu kurumlarına kadar birçok alanda kritik öneme sahiptir. Ancak, veritabanlarında oluşabilecek aykırılıklar (anomaliler), veri bütünlüğünü, performansı ve sistem güvenilirliğini tehdit edebilir. Geleneksel veritabanı yönetim sistemleri (DBMS), statik kural tabanlı yöntemlere dayanırken, dinamik ve değişen veri yapıları karşısında esneklikleri sınırlıdır.

Bu çalışmada, makine öğrenmesi ve derin öğrenmenin birlikte kullanıldığı hibrit bir sistem geliştirilerek, veritabanlarında otomatik anomali tespiti ve erken müdahale mekanizmaları

oluşturulması hedeflenmektedir.

- Makine öğrenmesi algoritmaları ile geçmiş verilere dayalı olarak veritabanı anormalliklerini öngörmek,
- Derin öğrenme modelleri ile karmaşık ve öngörülemeyen anomalileri daha yüksek doğrulukla tespit etmek,

▪ 1.1.2 Çalışmanın Önemi

Veritabanlarında meydana gelen anomaliler ve sistem hataları, sistem kesintilerine, veri kaybına ve güvenlik açıklarına yol açabilmektedir. Geleneksel veritabanı yönetim sistemleri, bu tür sorunları tespit etmek ve müdahale etmek için kural tabanlı yaklaşımlar ve manuel DBA (Database Administrator) müdahalelerine bağımlıdır. Ancak, günümüzün büyük ölçekli veri sistemleri için manuel yönetim süreçleri zaman alıcı ve yetersizdir.

Bu çalışmanın önemi aşağıdaki faktörlerle özetlenebilir:

1. Veritabanı sistemlerinde gerçek zamanlı anomali tespiti: Geleneksel sistemler genellikle anomalileri geç tespit eder ve çoğu zaman müdahale edemez. Bu sistem, otomatik ve hızlı bir şekilde anomalileri tespit edip müdahale etme yeteneğine sahip olacaktır.
2. Makine öğrenmesi ve derin öğrenme tekniklerinin hibrit kullanımı: Çalışma, ML tabanlı sınıflandırma modelleri ile DL tabanlı zaman serisi analiz yöntemlerini entegre ederek daha yüksek doğruluk oranına sahip bir tespit mekanizması sunacaktır.
3. Endüstri ve akademik alanda katkı sağlaması: Geliştirilecek sistem, kurumsal veritabanı yönetim sistemleri, finans, sağlık, e-ticaret ve büyük veri yönetimi gibi birçok alanda kullanılabilir. Ayrıca, yapay zeka destekli veritabanı yönetimi alanında akademik katkı sunacaktır.
4. Bu çalışmanın çıktıları, günümüzün büyüyen veri sistemlerinin daha güvenli, ölçeklenebilir ve otonom hale gelmesine katkı sağlayacak ve gelecekteki yapay zeka destekli veritabanı yönetim sistemlerine temel oluşturacaktır.

▪ 1.1.3 İlgili Çalışmalar

Veritabanı yönetim sistemlerinde aykırılık tespiti, hata tahmini ve optimizasyon konularında birçok çalışma yapılmış olup, bu çalışmalar genellikle makine öğrenmesi, derin öğrenme ve büyük dil modelleri (LLM) gibi yöntemleri içermektedir. Bu bölümde, önerilen çalışmaya benzer araştırmalar ele alınarak kullanılan yöntemler, temel bulgular ve eksik yönler tartışılacaktır.

Makale Başlığı	Yazar(lar)	Yayın Tarihi	Yayın Künyesi	Özet
D-Bot: Database Diagnosis System using Large Language Models [1]	Zhou, X., Li, G., Sun, Z., Liu, Z., Chen, W., Wu, J., ... & Zeng, G.	2023	Zhou, X., Li, G., Sun, Z., Liu, Z., Chen, W., Wu, J., ... & Zeng, G. (2023). D-bot: Database diagnosis system using large language models. arXiv preprint arXiv:2312.01454.	Bu çalışma, büyük dil modellerinin (LLM) veritabanı teşhis süreçlerinde kullanılmasını amaçlayan D-Bot adlı bir sistem önermektedir. D-Bot, otomatik teşhis ipuçları üreterek, kök neden analizleri yaparak ve işbirlikçi teşhis mekanizmaları kullanarak veritabanı yönetim süreçlerini optimize eder. Gerçek dünya verileriyle yapılan testler, D-Bot'un geleneksel DBA yöntemlerine kıyasla çok daha hızlı ve doğru sonuçlar verdiğini göstermektedir.
A Machine Learning Approach to Database Failure Prediction [2]	Karakurt, İ., Özer, S., Ulusinan, T., & Ganiz, M. C.	2017	Karakurt, İ., Özer, S., Ulusinan, T., & Ganiz, M. C. (2017). A machine learning approach to database failure prediction. In 2017	Bu çalışmada, Oracle veritabanı sistemlerindeki olası hataların önceden tahmin edilmesi için makine öğrenmesi

			International Conference on Computer Science and Engineering (UBMK) (pp. 1030-1035). IEEE.	teknikleri kullanılmıştır. Destek Vektör Makineleri (SVM), Naïve Bayes, Lojistik Regresyon, Random Forest ve AdaBoost gibi algoritmalar test edilmiş ve Random Forest %98.37 doğruluk ile en iyi performansı göstermiştir. Çalışma, veritabanı log verilerinden anormal olayları tahmin ederek şirketlerin arıza yönetimini iyileştirmesine katkı sağlamaktadır.
DB-GPT: Large Language Model Meets Database [3]	Zhou, X., Sun, Z., & Li, G.	2024	Zhou, X., Sun, Z., & Li, G. (2024). Db-gpt: Large language model meets database. Data Science and Engineering, 9(1), 102-111.	DB-GPT, büyük dil modellerinin veritabanı sorgu optimizasyonu, indeks ayarı ve sorgu yeniden yazımı gibi görevlerde nasıl kullanılabileceğini araştırmaktadır. GPT-3.5 tabanlı DB-GPT modeli, PostgreSQL'in yerleşik optimizasyon mekanizmasına kıyasla %22.4 daha iyi sorgu optimizasyonu ve %9.8 daha iyi indeks optimizasyonu

				sağlamıştır. Bu çalışma, LLM'lerin veritabanı yönetiminde nasıl kullanılabileceğini ortaya koyarak yeni nesil otomatik veritabanı optimizasyon sistemlerinin temelini atmaktadır.
Future Trends in SQL Databases and Big Data Analytics [4]	Islam, S.	2024	Islam, S. (2024). Future Trends In SQL Databases And Big Data Analytics: Impact of Machine Learning and Artificial Intelligence. Available at SSRN 5064781.	Bu makale, SQL veritabanları ile büyük veri analitiğinin kesişim noktalarını ve yapay zeka (AI) ile makine öğrenmesi (ML) tekniklerinin veri yönetimi süreçlerini nasıl dönüştürdüğünü incelemektedir. Makine öğrenmesi destekli sorgu optimizasyonu, AI tabanlı güvenlik sistemleri, SQL ve NoSQL hibrit entegrasyonları gibi konular ele alınmıştır. Çalışma, büyük veri çağında SQL sistemlerinin daha ölçeklenebilir ve verimli hale gelmesi için yeni teknolojilerin nasıl entegre edilebileceğini göstermektedir.
LLM for Data Management	Li, G., Zhou, X., & Zhao,	2024	Li, G., Zhou, X., & Zhao, X. (2024).	Bu çalışma, büyük dil modellerinin

[5]	X.		LLM for Data Management. Proceedings of the VLDB Endowment, 17(12), 4213-4216.	(LLM) veri yönetimi süreçlerinde nasıl kullanılabileceğini analiz etmektedir. Geleneksel makine öğrenmesi yöntemlerine kıyasla LLM'lerin bağlamsal anlama yetenekleri sayesinde veri analitiği, sorgu optimizasyonu ve hata teşhisi gibi alanlarda daha başarılı sonuçlar verebileceği belirtilmektedir. Önerilen sistem, RAG (Retrieval-Augmented Generation) teknikleri ve vektör veritabanları ile desteklenerek, LLM'lerin doğruluk oranını artırmayı hedeflemektedir.
Machine Learning for Database Management and Query Optimization [6]	Fahima, M. M. F., Sreen, A. S., Ruksana, S. F., Weihena, D. T. E., & Majid, M. H. M.	2024	Fahima, M. M. F., Sreen, A. S., Ruksana, S. F., Weihena, D. T. E., & Majid, M. H. M. (2024). Machine Learning for Database Management and Query Optimization. Elementaria: Journal of Educational Research, 2(1), 96-108.	Makine öğrenmesi tekniklerinin veritabanı yönetimi ve sorgu optimizasyonu alanında nasıl uygulanabileceğini araştıran bu çalışma, iş yükü yönetimi, indeksleme ve veri kalitesi sağlama gibi konulara odaklanmaktadır. Derin öğrenme, pekiştirmeli öğrenme ve doğal

				dil işleme tekniklerinin, veri tabanı yönetim sistemlerinde verimliliği artırma potansiyeli vurgulanmaktadır. Çalışma, özellikle dinamik sorgu optimizasyonu ve otonom indeksleme tekniklerine odaklanarak, geleneksel yöntemlerden daha iyi sonuçlar elde edilebileceğini göstermektedir.
--	--	--	--	--

Tablo 1.1 Literatür Tarama

1.2. Benchmark Çalışmaları

▪ 1.2.1. Patent Araştırması

Bu çalışmada, veri tabanı yönetimi ve yapay zeka konularında mevcut patentlerin varlığını araştırmak amacıyla çeşitli ulusal ve uluslararası patent veri tabanlarında aramalar gerçekleştirilmiştir. Araştırmada, aşağıdaki anahtar kelimeler kullanılmıştır:

- "database and deep learning"
- "database optimization"
- "database and machine learning"
- "database artificial intelligence"
- "veri tabanı ve yapay zeka"
- "veri tabanı"
- "veri tabanı ve derin öğrenme"
- "veri tabanı ve makine öğrenmesi"

Araştırma, ABD Patent ve Ticari Marka Ofisi (USPTO), Espacenet (Avrupa Patent Ofisi), Google Patents ve Türk Patent ve Marka Kurumu veri tabanlarında gerçekleştirilmiştir.

▪ 1.2.2. Patent Araştırma Sonuçları

USPTO **Patent** **Veri** **Tabanı:**

USPTO'nun Artificial Intelligence Patent Dataset (AIPD) 2023 güncellenmiş veri seti incelenmiş ve yukarıda belirtilen anahtar kelimelerle ilgili doğrudan bir patente rastlanmamıştır. Ancak, AI tabanlı veritabanı yönetimi ve optimizasyon süreçleri ile ilgili bazı genel patentler bulunmaktadır.

Espacenet **Patent** **Veri** **Tabanı:**

Espacenet üzerinden yapılan aramalarda, "database and deep learning", "database and machine learning" gibi anahtar kelimeler kullanılarak yapılan sorgular sonucunda doğrudan bir veritabanı ve yapay zeka entegrasyonu üzerine alınmış patent tespit edilememiştir.

Google **Patents** **Veri** **Tabanı:**

Google Patents üzerinde yapılan aramalarda, veritabanı optimizasyonu ve yapay zeka tabanlı veritabanı yönetimi ile ilgili bazı başvuruların olduğu görülmüş, ancak bu başvuruların doğrudan büyük dil modelleri (LLM) veya derin öğrenme tabanlı veritabanı teşhisi ve optimizasyonu ile ilgili olmadığı anlaşılmıştır.

Türk **Patent** **ve** **Marka** **Kurumu:**

Türk Patent ve Marka Kurumu veri tabanında "veri tabanı ve yapay zeka", "veri tabanı ve makine öğrenmesi" gibi Türkçe ve İngilizce anahtar kelimelerle araştırma yapılmış, ancak konuya ilişkin doğrudan bir patent bulunamamıştır.

▪ 1.2.3. Sonuç ve Değerlendirme

Yapılan benchmark çalışmaları sonucunda, veritabanı yönetim sistemleri ve yapay zeka entegrasyonu ile ilgili patentlerin sınırlı olduğu görülmektedir. Mevcut patentler genellikle genel veri tabanı optimizasyon teknikleri ve yapay zeka ile ilgili genel çözümler üzerine yoğunlaşmıştır. Derin öğrenme (deep learning) ve büyük dil modelleri (LLM) ile veritabanı teşhisi, optimizasyonu ve yönetimi konularında henüz kapsamlı bir patent başvurusunun olmadığı tespit edilmiştir.

2. MATERYAL VE YÖNTEM

2.1. Veri Seti ve Özellikleri

Bu çalışmada kullanılan veri seti, DataPlatform şirketinin gerçek verilerini içermektedir ve veritabanı performans analizine yönelik çeşitli metrikleri kapsamaktadır. Veri seti, zaman serisi tabanlı ölçümler içerdiğinden, veritabanı yönetimi süreçlerinin performansını değerlendirmek ve optimizasyon stratejileri geliştirmek için analiz edilmiştir.

▪ 2.1.1. Veri Setinin İçeriği

Veri setinde 36 farklı değişken bulunmaktadır. Bu değişkenler, veritabanının çalışma durumu, performans metrikleri ve yönetim süreçleri ile ilgili bilgiler içermektedir. Öne çıkan bazı değişkenler şunlardır:

- **InstanceID:** Veritabanı örneğini (instance) tanımlayan benzersiz kimlik numarasıdır.
- **CreateDate:** Verinin oluşturulma tarih ve saat bilgisi olup, zaman serisi analizleri için temel değişkendir.
- **page life expectancy:** Veritabanı sayfalarının bellekte ne kadar süre tutulduğunu belirten metrik olup, bellek yönetimi optimizasyonu için önemlidir.
- **% privileged time:** CPU'nun veritabanı işlemleri için ayırdığı öncelikli süreyi gösteren performans metriğidir.
- **transactions/sec:** Veritabanında gerçekleşen işlemlerin (transactions) saniye başına oranını gösterir.
- **write transactions/sec:** Yazma işlemleri için saniye başına gerçekleşen transaction sayısını belirtir.
- **logical connections:** Veritabanına bağlı olan istemcilerin sayısını gösteren metrik olup, yük yönetimi ve ölçeklenebilirlik analizlerinde önemli bir parametredir.
- **dbaHCDOC, dbaShrinkLog, dbaFullAylıkBackup:** Veritabanı yöneticileri tarafından gerçekleştirilen bakım işlemlerini ifade eden değişkenlerdir.
- **syspolicy_purge_history:** Sistem politikalarına göre gereksiz verileri temizleme işlemlerini gösterir.

▪ 2.1.2. Veri Ön İşleme ve Temizleme

Veri seti üzerinde eksik verilerin doldurulması, aykırı değerlerin temizlenmesi ve normalizasyon işlemleri gerçekleştirilmiştir.

▪ 2.1.3. Veri Setinin Kullanım Amacı

Bu veri seti, aşağıdaki amaçlarla analiz edilmiştir:

1. Veritabanı Performans Analizi:

- İşlem yükü (transactions/sec) ve CPU kullanım oranlarının korelasyonu analiz edilerek, sistem üzerindeki yoğunluk dönemleri belirlenmiştir.
- Page Life Expectancy değişkeni ile bellek kullanım verimliliği değerlendirilmiştir.

2. Anomali Tespiti:

- Normalden sapmalar gösteren CPU ve bellek kullanımları tespit edilerek, sistem anormalliklerini belirlemek için Makine Öğrenmesi tabanlı anomali tespiti uygulanmıştır.

▪ 2.1.4 Veri Seti Örneği

InstanceID	CreateDate	page life expectancy	% privileged time	transactions/sec	write transactions/sec	logical connections	dbaKDOC	VobVadesiDolmuslarisili	dbaShrinklog	...	syspolicy_purge_history	dbaCheckinternerStatus	SP_tblTicunlukInsertData	tblTicFarihnel
0	297 2024-06-20 00:00:00	12167	10.15	254.72	166.82	638	0	0	0	...	0	0	0	0
1	297 2024-06-20 00:00:15	12167	3.32	420.75	336.80	658	0	0	0	...	0	0	0	0
2	297 2024-06-20 00:00:30	12167	2.54	217.77	174.81	655	0	0	0	...	0	0	0	0
3	297 2024-06-20 00:00:45	12167	4.09	159.67	124.74	652	0	0	0	...	0	0	0	0
4	297 2024-06-20 00:01:00	12167	3.12	411.09	190.58	652	0	0	0	...	0	0	0	0

5 rows * 36 columns

Şekil 2.1: Veri Setinin İlk Beş Satırı

2.2. Makine Öğrenmesi ve Derin Öğrenme Algoritmaları

Bu çalışmada, veritabanı performans analizi, anomali tespiti ve tahmine dayalı modelleme süreçlerinde farklı makine öğrenmesi ve derin öğrenme algoritmaları kullanılmıştır. Veri setindeki örüntüleri belirlemek, veri tabanı üzerindeki işlemleri optimize etmek ve anomali tespiti yapmak amacıyla aşağıdaki algoritmalar tercih edilmiştir.

▪ 2.2.1 Makine Öğrenmesi Algoritmaları ve Hiperparametreler

Bu çalışmada, verilerin sınıflandırılması ve tahmin edilmesi amacıyla üç farklı makine öğrenmesi algoritması uygulanmıştır. Bu algoritmaların her birine ilişkin kullanılan hiperparametreler aşağıda detaylı olarak sunulmuştur.

• 1. Random Forest Algoritması:

Random Forest, birden fazla karar ağacının birleşiminden oluşan bir öğrenme yöntemidir. Bu algoritma, güçlü tahmin performansı ile bilinmekte olup, yüksek doğruluk oranlarına ulaşabilmektedir. Random Forest algoritmasında kullanılan belirli hiperparametreler şunlardır:

- **n_estimators:** 100
Modeldeki karar ağaçlarının sayısı belirlenmiştir. 100, makul bir model doğruluğu elde etmek için seçilen optimum değeri temsil etmektedir.
- **max_depth:** 5
Karar ağaçlarının maksimum derinliği sınırlandırılmıştır. Bu, modelin aşırı öğrenmesini (overfitting) engellemek için kullanılan bir parametredir.
- **min_samples_split:** 2
Bir iç düğümün iki alt düğüme ayrılması için gereken minimum örnek sayısını ifade etmektedir. Bu parametre, modelin daha ince ayrıntılara odaklanmasına olanak verir.
- **criterion:** Gini
Karar ağacı inşasında, düğüm bölme kararlarını almak için kullanılan kriterdir. Burada, Gini saflığı kullanılmıştır.
- **Test Size:** 0.2
Verilerin %20'lik kısmı test seti olarak ayrılmıştır.
- **Random State:** 42
Rastgeleliğin tekrarlanabilirliğini sağlamak amacıyla belirlenen sabit bir sayı.

• 2. SVM (Support Vector Machine) Algoritması:

Destek Vektör Makinesi (SVM), verilerin en iyi şekilde sınıflandırılabilmesi için bir hiper düzlem (hyperplane) bulmaya çalışan bir algoritmadır. SVM algoritmasında kullanılan

hiperparametreler ise şu şekildedir:

- **C:** 2.00
C parametresi, modelin sınıflandırma hatalarını nasıl cezalandıracağını belirler. Daha büyük C değerleri, modelin hata toleransını azaltır.
- **Kernel:** Sigmoid
SVM'nin doğrusal olmayan sınıflandırmalar için kullanılan çekirdek fonksiyonudur. Burada, sigmoid fonksiyonu tercih edilmiştir.
- **Sütun Aralığı:** 2 (page life expectancy) - 7 (dbaHCDOC)
Kullanılan özellikler arasındaki ilişkiyi tanımlar. Bu aralık, SVM modeline sağlanan veri sütunlarını belirlemektedir.
- **Test Size:** 0.2
Test verisinin oranını belirler. Verilerin %20'lik kısmı test için ayrılmıştır.
- **Random State:** 42
Modelin çalıştırılabilirliğini ve rastgeleliğinin sabitliğini sağlamak amacıyla belirlenen değer.
- **Operator:** >
Sınıflandırma için kullanılan karşılaştırma operatörüdür.
- **Hedef Etiket:** 300
Sınıflandırılacak hedef etiket değeri.
- **Hedef Etiket Tanımı:** $y = df["page\ life\ expectancy"] > 300$
Hedef etiket, verinin "page life expectancy" sütununun 300'den büyük olup olmadığına göre belirlenmiştir.
- **3. KNN (K-Nearest Neighbors) Algoritması:**

K-Nearest Neighbors (KNN) algoritması, en yakın komşulara dayalı bir sınıflandırma yöntemidir. KNN, etiketlerin öğrenilmesinde yakınlık ölçütlerini kullanır. Kullanılan hiperparametreler aşağıda verilmiştir:

- **n_neighbors:** 5
KNN modelinde, sınıflandırma kararı verirken dikkate alınacak komşu sayısını belirler.

- **metric:** Euclidean
Komşuluk mesafesinin hesaplanmasında kullanılan metriktir. Burada, Öklid mesafesi tercih edilmiştir.
- **weights:** Uniform
Komşu ağırlıklarının tüm komşular için eşit olduğu belirtilmiştir. Bu, her komşunun sınıf kararına eşit katkı sağlamasını sağlar.
- **Sütun Aralığı:** 2 (page life expectancy) - 7 (dbaHCDOC)
KNN modeline sağlanan veri sütunlarının seçildiği aralık.
- **Test Size:** 0.2
Test verisinin oranını belirler. Verilerin %20'si test için ayrılmıştır.
- **Random State:** 42
Modelin rastgeleliğini kontrol altına almak için sabit bir değer belirlenmiştir.
- **Operator:** >
KNN algoritmasında kullanılan karşılaştırma operatörüdür.
- **Hedef Etiket:** 300
KNN algoritmasında sınıflandırılması gereken hedef etiket değeri.
- **Hedef Etiket Tanımı:** $y = df["page\ life\ expectancy"] > 300$
Hedef etiket, verinin "page life expectancy" sütununun 300'den büyük olup olmadığına göre belirlenmiştir.

2.2.2 Kullanılan Makine Öğrenmesi Sonuçları

Algorithm	Confusion Matrix	Accuracy	F1-Score	Recall
SVM	[[1864, 137], [104, 11889]]	0.9828	0.9900	0.9913
Random Forest	[[1593, 408, 0], [0, 5482, 0], [0, 0, 6511]]	0.9708	0.9708	0.9708
KNN	[[2004, 0], [25, 1978]]	0.9938	0.9937	0.9875

Tablo 2.1 Üç Algoritmaya Ait Model Eğitim Sonuçları

- **Tablo 2.1'e Ait Açıklamalar:**
- **Confusion Matrix:** Her algoritmanın, doğru ve yanlış sınıflandırdığı örneklerin sayısını göstermektedir. Satırlar gerçek sınıfı, sütunlar ise tahmin edilen sınıfı ifade eder. Örneğin, SVM algoritması için [1864, 137] ilk satırda, [104, 11889] ikinci satırda yer alıyor, bu da doğru ve yanlış sınıflandırmalar hakkında bilgi verir.
- **Accuracy (Doğruluk):** Modelin doğru tahminlerinin, toplam tahminlere oranıdır. Yüksek bir accuracy, modelin genel doğruluğunu gösterir. KNN algoritması, en yüksek doğruluğa sahip olup, %99.38 doğruluk oranına ulaşmıştır.
- **F1-Score:** Precision ve Recall değerlerinin harmonik ortalamasıdır. F1-Score, dengesiz sınıf dağılımlarında daha faydalıdır. SVM ve KNN algoritmaları, yüksek F1-Score değerleri ile dikkat çekmektedir.
- **Recall:** Gerçek pozitif sınıfların doğru bir şekilde sınıflandırılma oranını ifade eder. Yüksek bir recall değeri, modelin doğru pozitif sınıflandırmalarını artırma yeteneğini gösterir. SVM, en yüksek recall değerine sahip algoritmadır.

▪ 2.2.3 Kullanılması Planlanan Derin Öğrenme Algoritmaları

Veritabanı yönetimi ve performans optimizasyonunda zaman serisi analizi ve tahminleme için derin öğrenme modellerin kullanılması hedeflenmektedir.

- **Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNN):**
 - Önceki zaman adımlarını dikkate alarak ardışık verilerde analiz yapar.
 - Veritabanındaki işlem yükü değişikliklerini zaman serisi bazında analiz etmek için kullanılmıştır.
 - Ani performans değişikliklerini öngörerek gelecekteki anormallikleri tespit edebilme yeteneğine sahiptir.
- **Uzun Kısa Süreli Bellek (Long Short-Term Memory - LSTM):**
 - RNN modelinin geliştirilmiş versiyonudur ve uzun vadeli bağımlılıkları öğrenme kapasitesine sahiptir.
 - Veritabanı işlemleri üzerindeki zaman serisi tahminlerinde ve performans optimizasyonunda kullanılmıştır.

- Bellek mekanizması sayesinde geçmiş işlemleri daha etkili bir şekilde hatırlayarak gelecek tahminleri yapabilmektedir.

3. WEB SİTESİ GELİŞTİRME VE KULLANILAN TEKNOLOJİLER

Bu çalışmada, yapay zeka destekli bir web sitesi geliştirilerek, kullanıcıların veri analizi yapmasına, makine öğrenmesi modelleri eğitmesine ve sonuçları raporlamasına olanak sağlayan bir platform oluşturulmuştur. Web sitesi, React ve Flask gibi modern teknolojileri kullanarak veritabanı yönetimi, analiz, model eğitimi ve raporlama işlemlerini entegre eden bir sistem olarak tasarlanmıştır.

3.1. Web Sitesinin Genel Yapısı

Web sitesi, kullanıcı dostu bir arayüze sahip olup, veri yükleme, model seçimi, eğitim süreci, sonuçların görüntülenmesi ve raporlanması gibi adımları içermektedir. Platformun temel özellikleri şu şekildedir:

- **Kullanıcı Girişi ve Model Eğitimi:** Kullanıcılar sisteme giriş yaptıktan sonra, makine öğrenmesi veya derin öğrenme modellerini eğitmek için veri yükleyebilir. Model eğitimi tamamlandığında, kullanıcılar görselleştirilmiş sonuçları analiz edebilir ve seçtikleri metrikler doğrultusunda şirket kategorisini belirleyebilir. Bu özellik tamamlanmış olup, kullanıcı deneyimini iyileştirmek için daha fazla model seçeneği eklenmesi planlanmaktadır.
- **Dosya Yükleme ve İşleme:** Kullanıcılar veri setlerini yükleyerek model eğitimine başlayabilir ve arka planda çalışan algoritmalar sayesinde veri ön işleme adımlarını gerçekleştirebilir. Eğer yüklenen veri, sistemdeki mevcut veriyle uyumluysa, otomatik birleştirme işlemi yapılabilir. Bu özellik tamamlanmış olup, veri boyutları arttıkça işlem süresi iyileştirilecek ve optimizasyon yapılacaktır.
- **Gerçek Zamanlı Sonuç Analizi:** Yapay zeka destekli anlık yorumlama özelliği sayesinde, model eğitimi sonucunda oluşan grafikler ChatGPT tarafından analiz edilerek kullanıcıya detaylı açıklamalar sunulmaktadır. Bu özellik tamamlanmış olup, kullanıcı deneyimini geliştirmek için analizlerde daha fazla kişiselleştirme ve özelleştirme seçenekleri eklenmesi

planlanmaktadır.

- **Raporlama ve PDF Oluşturma:**

Kullanıcılar, eğitim sonuçlarını PDF formatında rapor olarak kaydedebilir ve her oluşturulan rapor, kullanıcının hesabına otomatik olarak kayıt edilmektedir. Bu özellik yapılacaktır. Kullanıcılar için daha fazla raporlama seçeneği ve kişiselleştirilebilir rapor formatları eklenmesi planlanmaktadır.

- **Telegram Bot Bildirim Sistemi:**

Model eğitimi sırasında, Telegram botu üzerinden anlık doğruluk (accuracy) değeri bildirimleri gönderilmekte, böylece kullanıcı modelin ilerleme durumunu gerçek zamanlı olarak takip edebilmektedir. Bu özellik yapılacaktır. Telegram botu üzerinden ek bildirim türleri ve özelleştirilebilir bildirim seçenekleri eklenmesi planlanmaktadır.

- **Admin Paneli ve Yönetim:**

Web sitesi üzerinde bir admin paneli oluşturulmuş olup, admin kullanıcıları diğer kullanıcıların gerçekleştirdiği model eğitimlerini, raporlamalarını ve genel sistem istatistiklerini takip edebilmektedir. Bu özellik yapılacaktır. Admin paneline ek yönetim araçları ve kullanıcı raporlama seçenekleri eklenmesi planlanmaktadır.

3.2. Kullanılan Teknolojiler ve Araçlar

Web sitesinin geliştirilme sürecinde, modern frontend ve backend teknolojileri kullanılarak kullanıcı dostu, ölçeklenebilir ve yüksek performanslı bir sistem oluşturulmuştur. Kullanılan temel araçlar şunlardır:

- **Backend (Sunucu Tarafı)**
- **Python & Flask:**
 - Web API'lerinin geliştirilmesi, veritabanı yönetimi ve model eğitimi süreçleri için kullanılan hafif ve ölçeklenebilir bir Python web çerçevesidir.
 - Kullanıcı doğrulama, veri yükleme, model eğitimi ve sonuç raporlamalarını yönetmek için Flask ile RESTful API'ler oluşturulmuştur.
- **MSSQL Server:**
 - Veri yönetimi ve saklama süreçleri için kullanılan ilişkisel veritabanı yönetim sistemi (RDBMS).

- Kullanıcı işlemleri, model eğitim sonuçları ve analiz verileri MSSQL üzerinde saklanmaktadır.
- **Frontend (Kullanıcı Arayüzü - UI/UX)**
- **React.js:**
 - Kullanıcıların etkileşimli ve dinamik bir deneyim yaşamasını sağlamak için bileşen tabanlı modern JavaScript kütüphanesi kullanılmıştır.
 - Veri yükleme, model seçimi, eğitim süreci ve sonuçların görselleştirilmesi React bileşenleri aracılığıyla yönetilmektedir.
- **Ant Design (AntD):**
 - Web arayüzünün modern ve profesyonel bir görünüme sahip olması için kullanılan React tabanlı UI bileşen kütüphanesidir.
 - Kullanıcı girişi, veri yükleme, grafik gösterimi, butonlar ve form elemanları gibi bileşenler için Ant Design bileşenleri kullanılmıştır.
- **Bildirim ve Otomasyon Sistemleri**
- **Telegram Bot API:**
 - Model eğitimi sırasında doğruluk (accuracy) değeri ve diğer önemli istatistikler Telegram botu aracılığıyla kullanıcıya gerçek zamanlı olarak bildirilecektir.
 - Kullanıcılar, model eğitimi tamamlandığında Telegram üzerinden otomatik güncellemeler alabileceklerdir.
- **Raporlama ve Dosya Yönetimi**
- **FPDF (Python PDF Kütüphanesi):**
 - Kullanıcıların eğitim sonuçlarını ve analizlerini PDF formatında raporlamasına olanak sağlayan kütüphane kullanılacaktır.
 - Kullanıcı, sonuçları analiz ettikten sonra "Rapor Oluştur" butonuna tıklayarak PDF formatında çıktı alabilecektir.
 - FPDF ile oluşturulan PDF dosyaları, kullanıcının hesabına kaydedilecek ve istenildiğinde indirilebilecektir.

3.3. Web Sitesine Ait Görseller

Makine Öğrenme Algoritmaları

✓ Birinci Adım 2 İkinci Adım 3 Son Adım

Eğitim için İstedğiniz Dosya/Dosyaları Yükleyiniz.

⬇ Kullanım Kılavuzu İndir ⬇ CSV Oluşturma İçin Scripti İndir

⬇ Sadece CSV Yükle

Algoritma Seçimi Yap

Yüklenen Dosya Sayısı: 0

Sonraki Adım Önceki Adım

Şekil 3.1: Yüklenmesi Gereken Veri Seti Dosyası

Makine Öğrenme Algoritmaları

✓ Birinci Adım 2 İkinci Adım 3 Son Adım

Eğitim için İstedğiniz Dosya/Dosyaları Yükleyiniz.

⬇ Kullanım Kılavuzu İndir ⬇ CSV Oluşturma İçin Scripti İndir

⬇ Sadece CSV Yükle

OYAKint2.csv

Algoritma Seçimi Yap

Yüklenen Dosya Sayısı: 1

☐ SVM Algoritması ☐ Random Forest Algoritması ☐ KNN Algoritması ☐ Karşılaştırmalı Algoritma

Sonraki Adım Önceki Adım

Şekil 3.2 : Algoritma Seçimi

☐ SVM Algoritması ☒ Random Forest Algoritması ☐ KNN Algoritması ☐ Karşılaştırmalı Algoritma

Şirketinizi Seçiniz: Seçim için arama yapab Random Forest, karmaşık veri setlerinde yüksek performans gösterir ve genellikle overfiteeğe karşı dirençlidir. Aynı zamanda, çok sayıda girdi değişkeniyle çalışırken de etkili sonuçlar verir. Algoritma, her bir ağacı farklı bir alt küme veriyle eğittiği için varyansı azaltır ve genel olarak iyi genelleme yapabilir. Ayrıca, Random Forest, özelliklerin önem sıralamasını elde etmek için kullanışlı olabilir ve aşırı uç değerlere karşı dirençli olma özelliğiyle de dikkat çeker. Bu özellikleriyle, Random Forest genellikle sınıflandırma ve regresyon problemlerinde tercih edilen bir algoritma haline gelmiştir.

n_estimators:

max_depth:

min_samples_split:

criterion:

Test Size:

Random State:

Varsayılan Değeri Seç

Eğitimi Tamamla

Şekil 3.3 : Veri Ön İşleme Aşamasında Hiperparametre Seçimi

4. BULGULAR VE TARTIŞMA

4.1. Şu Ana Kadar Yapılan Çalışmalar

4.1.1 Veri Toplama ve Ön İşleme

- DataPlatform şirketine ait gerçek veriler kullanılmış ve veri tabanı yönetimi, performans analizi ve anomali tespiti için uygun hale getirilmiştir.
- Eksik veriler doldurulmuş, aykırı değerler temizlenmiş ve veri seti analiz için optimize edilmiştir.

4.1.2 Web Sitesinin Geliştirilmesi

- React.js ve Ant Design kullanılarak kullanıcı dostu ve etkileşimli bir arayüz oluşturulmuştur.
- Flask & Python kullanılarak backend API'leri geliştirilmiş ve MSSQL veritabanı ile entegre edilmiştir.

4.1.3 Kullanılan Makine Öğrenmesi Modelleri

- Random Forest, KNN, SVM gibi klasik makine öğrenmesi modelleri kullanılarak anomali tespiti ve performans analizleri yapılmıştır.

▪ 4.1.4 Model Performans Değerlendirmesi

- Eğitim sürecinde doğruluk (accuracy), kesinlik (precision), geri çağırma (recall) ve F1 skoru gibi metrikler kullanılarak model başarıları analiz edilmiştir.
- Makine öğrenmesi algoritmalarının karşılaştırmalı analizleri yapılmış ve hangi modelin hangi senaryolarda daha başarılı olduğu değerlendirilmiştir.

4.2. Gelecekte Yapılacak Çalışmalar

▪ 4.2.1 Kullanılması Planlanan Modeller

RNN (Recurrent Neural Networks) ve LSTM (Long Short-Term Memory) gibi derin öğrenme tabanlı modellerle zaman serisi tahminleri yapılması planlanmaktadır. Bu modeller, veritabanı optimizasyonu için geleceğe yönelik öngörülerde bulunmak amacıyla kullanılacak ve sistemin, ilerleyen dönemdeki veri talepleriyle uyumlu hale gelmesi sağlanacaktır.

▪ 4.2.2 Web Sitesine Yeni Özellikler Eklenmesi

- **Otomatik Model Seçimi (AutoML):** Kullanıcıların manuel parametre seçimi yapmalarına gerek kalmadan en uygun modelin otomatik olarak belirlenmesini sağlayan "Otomatik Model Seçimi" özelliği web sitesine entegre edilecektir.
- **Telegram Bot API Entegrasyonu:** Model eğitim süreçlerinin sonuçları, Telegram Bot API aracılığıyla kullanıcıya anlık olarak bildirilecektir. Bu bildirimler, doğruluk (accuracy) oranı ve diğer kritik istatistikleri içerecek, kullanıcıların model performansını gerçek zamanlı olarak takip etmelerini sağlayacaktır.
- **FPDF Kütüphanesi ile Otomatik Raporlama:** FPDF Python kütüphanesi kullanılarak, eğitim sonuçlarının ve analizlerin PDF formatında otomatik olarak raporlanması sağlanacaktır. Bu sistem, kullanıcıların raporlama süreçlerini daha verimli ve kullanıcı dostu bir şekilde gerçekleştirmelerine olanak tanıyacaktır.

5. SONUÇ

Yapılan çalışmalar, veritabanı yönetimi süreçlerinin yapay zeka ile nasıl optimize edilebileceğini göstermektedir. Web tabanlı sistemimiz, kullanıcıların otomatik model eğitimi yapmasını, anomali tespit etmesini ve performans analizleri gerçekleştirmesini

sağlayan güçlü bir platform sunmaktadır.

Gelecekte, daha büyük veri kümeleri üzerinde test edilerek model doğruluğu artırılacak, sistemin ölçeklenebilirliği iyileştirilecek ve kullanıcı deneyimi geliştirilerek tam otomatik bir analiz platformu oluşturulacaktır.

KAYNAKLAR

- [1] X. Zhou et al., “D-Bot: Database Diagnosis System using Large Language Models,” arXiv.org, 2024. <https://arxiv.org/abs/2312.01454>
- [2] Ismet Karakurt, S. Ozer, Taner Ulusinan, and Murat Can Ganiz, “A machine learning approach to database failure prediction,” 2017 International Conference on Computer Science and Engineering (UBMK), pp. 1030–1035, Oct. 2017, doi: <https://doi.org/10.1109/ubmk.2017.8093426>.
- [3] X. Zhou, Z. Sun, and G. Li, “DB-GPT: Large Language Model Meets Database,” Data science and engineering (Internet), Jan. 2024, doi: <https://doi.org/10.1007/s41019-023-00235-6>.
- [4] S. Islam, “FUTURE TRENDS IN SQL DATABASES AND BIG DATA ANALYTICS: IMPACT OF MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE,” vol. 1, no. 4, pp. 47–62, Aug. 2024, doi: <https://doi.org/10.62304/ijse.v1i04.188>.
- [5] G. Li, X. Zhou, and X. Zhao, “LLM for Data Management,” Proceedings of the VLDB Endowment, vol. 17, no. 12, pp. 4213–4216, Aug. 2024, doi: <https://doi.org/10.14778/3685800.3685838>.
- [6] “View of Machine Learning for Database Management and Query Optimization,” Elementaria.my.id, 2024. <https://elementaria.my.id/index.php/e/article/view/66/23>