

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MERKEZLERİ İÇİN MAKİNE ÖĞRENMESİ TEMELLİ
İZLEME SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Metehan HAN

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

ŞUBAT 2025

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

VERİ MERKEZLERİ İÇİN MAKİNE ÖĞRENMESİ TEMELLİ
İZLEME SİSTEMİ TASARIMI

YÜKSEK LİSANS TEZİ

Metehan HAN

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Bilim Dalı

Tez Danışmanı: Dr.Öğr.Üyesi Abdullah SEVİN

ŞUBAT 2025

Metehan HAN tarafından hazırlanan “Veri Merkezleri için Makine Öğrenmesi Temelli İzleme Sistemi Tasarımı” adlı tez çalışması 05.02.2025 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı **Bilgisayar Mühendisliği** Bilim Dalı’nda Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :

Jüri Üyesi :

Jüri Üyesi :



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum “Veri Merkezleri için Makine Öğrenmesi Temelli İzleme Sistemi Tasarımı” başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi’nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim.

(05/02/2025)

Metehan HAN





Eşime ve Bilime.



TEŞEKKÜR

Bu tezin hazırlanması sürecinde, destek ve katkılarıyla bana yol gösteren birçok kişi ve kuruma teşekkür etmek isterim. Öncelikle, tez danışmanım Dr. Öğr. Üyesi Abdullah SEVİN'e, değerli görüşleri ve sürekli desteği için en içten teşekkürlerimi sunarım. Araştırmamın her aşamasında bana rehberlik ederek, düşüncelerimi şekillendirmemde büyük rol oynadınız.

Ayrıca, veri merkezleri ve makine öğrenmesi alanındaki bilgi ve deneyimlerini paylaşan tüm akademisyen ve uzmanlara teşekkür ediyorum. Sunduğunuz kaynaklar ve öneriler, çalışmamı zenginleştirdi. Bu süreçte yanımda olan aileme, arkadaşlarıma ve tüm destekçilerime de teşekkür etmek isterim. Sabrınız ve cesaretlendirmeniz benim için büyük bir motivasyon kaynağı oldu.

Son olarak, bu çalışmanın gerçekleştirilmesine olanak tanıyan Sakarya Üniversitesi'ne, sağladığınız kaynaklar ve imkanlar için minnettarım.

Metehan HAN



İÇİNDEKİLER

Sayfa

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	v
TEŞEKKÜR	ix
İÇİNDEKİLER	xi
KISALTMALAR	xv
SİMGELER	xvii
TABLO LİSTESİ	xix
ŞEKİL LİSTESİ	xxi
ÖZET	xxiii
SUMMARY	xxv
1. GİRİŞ.....	1
1.1. Motivasyon.....	2
1.2. Problem Tanımı.....	3
1.3. Tezin Kapsamı.....	3
1.4. Tezin Amacı	4
1.5. Yöntem	4
1.6. Bilime Katkısı	5
2. LİTERATÜR ARAŞTIRMASI	7
2.1. Veri Merkezi İzleme Sistemleri	9
2.2. Makine Öğrenmesi ve Veri Merkezi Yönetimi	10
2.3. Bulut Sistemlerde İzleme	11
2.4. Öngörücü İzleme ve Proaktif Yönetim	11
3. YÖNTEM.....	13
3.1. Örneklem	13
3.2. Veri Toplama ve Entegrasyon	14
3.2.1. Veri toplama nedir?	14
3.2.2. Veri entegrasyonu nedir?	15
3.2.3. Makine öğrenmesi nedir?	15
3.2.4. Makine öğrenmesi modelleri ve izleme uygulamaları	16
3.2.4.1. Denetimli öğrenme (supervised learning)	16
3.2.4.2. Denetimsiz öğrenme (unsupervised learning)	16
3.2.4.3. Takviyeli öğrenme (reinforcement learning)	16
3.2.4.4. Derin öğrenme (deep learning)	17
3.2.5. Makine öğrenmesinin izleme sistemlerindeki yeri	17
3.3. Trend Verilerine Dayalı İzleme Teknikleri	17
3.3.1. Trend verileri nedir?	17
3.3.2. Trend izleme teknikleri	18
3.3.2.1. Zaman serisi analizi	18
3.3.2.2. Tahmin modelleri ve kapasite planlama	18
3.3.3. İzleme, veri görselleştirme ve alarm sistemleri	19
3.4. Veri Merkezi İzlemelerinde Kullanılan Uygulamalar	19
3.4.1. Prometheus ve grafana	19

3.4.2. Elastic stack (ELK)	20
3.4.3. Zabbix.....	20
3.4.4. Dynatrace	21
3.4.5. New relic	22
3.4.6. SigNoz ve openTelemetry	23
3.4.7. Nagios.....	23
3.4.8. Datadog	24
3.4.9. Arbor edge defense (AED).....	25
3.4.10. Amazon cloudwatch.....	26
3.5. Geleneksel İzleme Sistemleri	26
3.6. Makine Öğrenimi (ML) Tabanlı İzleme.....	27
3.7. Yapay Zeka (AI) Tabanlı İzleme.....	27
4. SİSTEM TASARIMI VE UYGULAMA.....	29
4.1. Alarm Sistemi.....	30
4.2. Diğer Sistemlerle Entegrasyon	31
4.3. Makine Öğrenimi.....	31
4.4. Genel Tasarım Yaklaşımı	33
4.4.1. Sistemin amaçları ve gereksinimleri	33
4.4.2. Rol tabanlı mimari.....	34
4.4.3. Ölçeklenebilirlik.....	34
4.4.4. Gelişmiş entegrasyon ve uzun vadeli uyumluluk.....	34
4.5. Makine Öğrenmesi Entegrasyonu	35
4.5.1. Dahili trend analizi ve tahminsel izleme	35
4.5.2. Baseline oluşturma ve analizi.....	36
4.5.3. Dinamik eşik ayarlama ve analizi	37
4.5.4. Harici makine öğrenimi modelleri ile entegrasyon	38
4.6. Entegrasyon ve Test Süreçleri	38
4.7. Veri Toplama ve Entegrasyon	38
4.8. Makine Öğrenmesi Modelleri.....	39
4.9. Trend Verilerine Dayalı İzleme Teknikleri	39
4.9.1. Çalışma prensibi.....	39
4.9.2. Algoritma detayları	40
4.9.2.1. Deviations (sapmalar)	42
4.10. Baseline' a Dayalı İzleme Teknikleri	44
4.10.1. Çalışma prensibi.....	44
4.10.2. Algoritma detayları	45
4.11. İleriye Yönelik Tahmine Dayalı İzleme Teknikleri	46
4.11.1. Çalışma prensibi.....	46
4.11.2. Algoritma detayları	47
4.12. Otomatik İzleme ve Uyarı Sistemleri	48
4.13. Deneysel Ortam ve Veri Seti	49
4.14. Uygulama Mimarisi.....	49
4.14.1. Zabbix server uygulaması	50
4.14.2. Zabbix agent uygulaması	50
4.14.3. Veri toplama servisi	50
4.14.4. Ön işleme ve değerlendirme.....	51
4.14.5. Uyarı ve bildirim sistemi.....	51
5. UYGULAMA VE TEST	53
6. SONUÇLAR VE TARTIŞMA	71
KAYNAKLAR.....	73

ÖZGEÇMİŞ.....	77
---------------	----





KISALTMALAR

AED	:Arbor Edge Defense (Arbor Sınır Savunması)
AI	:Artificial intelligence (Yapay Zeka)
AIOPS	:Artificial Intelligence for IT Operations (Yapay Zeka BT Operasyonları)
ANNs	:Artificial Neural Networks (Yapay Sinir Ağları)
API	:Application Programming Interface (Uygulama Programlama Arayüzü)
APM	:Application Performance Monitoring (Uygulama Performansı İzleme)
ARIMA	:Autoregressive Integrated Moving Average (Oto regresif Entegre Hareketli Ortalama)
AWS	:Amazon Web Services (Amazon Web Hizmetleri)
CPU	:Microsoft SQL Server (Microsoft SQL Sunucusu)
DCN	:Dynamic Circuit Network (Dinamik Devre Ağı)
DDoS	:Distributed Denial of Service (Dağıtılmış Hizmet Reddi)
EC2	:Elastic Compute Cloud (Elastic Bilgi İşlem Bulutu)
ELK	:Elasticsearch, Logstash and Kibana
I/O	:Input/Output (Giriş/Çıkış)
ICMP	:Internet Control Message Protocol (İnternet Kontrol Mesaj Protokolü)
IOT	:Internet of Things (Nesnelerin İnterneti)
IT	:Information Technologies (Bilgi Teknolojileri)
ITSM	:Information Technology Service Management (Bilgi Teknolojileri Servis Yönetimi)
JMX	:Java Management Extensions (Java Yönetim Uzantıları)
LOESS	:Locally estimated scatterplot smoothing (Yerel olarak tahmin edilen dağılım grafiği yumuşatma)
LSTM	:Long Short-Term Memory (Uzun Kısa Süreli Bellek)
ML	:Machine Learning (Makine Öğrenmesi)
MSSQL	:Microsoft SQL Server (Microsoft SQL Sunucusu)
MySQL	:My Structured Query Language (Yapılandırılmış Sorgu Dili)
PostgreSQL	:Postgre Structured Query Language (Postgre Yapılandırılmış Sorgu Dili)
PromQL	:Prometheus Query Language (Prometheus Sorgu Dili)

REST API	:Representational State Transfer Application Programming Interface (Temsili Durum Transferi Uygulama Programlama Arayüzü)
RNN	:Recurrent Neural Networks (Tekrarlayan Sinir Ağları)
SDK	:Software Development Kit (Yazılım Geliştirme Kiti)
SMS	:Short Message Service (Kısa Mesaj Servisi)
SNMP	:Simple Network Management Protocol (Basit Ağ Yönetim Protokolü)
SQL	:Structured Query Language (Yapılandırılmış Sorgu Dili)
STL	:Seasonal and Trend decomposition using Loess (Loess kullanarak Mevsimsel ve Trend ayrıştırması)
SVM	:Support Vector Machine (Destek Vektör Makinesi)
WMA	:Weighted Moving Average (Ağırlıklı Hareketli Ortalama)
WMI	:Windows Management Instrumentation (Windows Yönetim Araçları)



SİMGELER

σ	: Standart Sapma
%	: Yüzde
Σ	: Toplama Operatörü





TABLO LİSTESİ

Sayfa

Tablo 4.1. Geleneksel ve ML Tabanlı İzleme Karşılaştırması.....	33
Tablo 4.2. Timeleft ve Forecast Karşılaştırması:	48
Tablo 5.1. Zabbix İtem Veri İzleme Tanımları	55
Tablo 5.2. Zabbix Alarm Tanımları	57
Tablo 5.3. Geleneksel İzleme ile Makine Öğrenmesi İzlemesinin Karşılaştırılması	68
Tablo 5.4. Geleneksel İzleme ile Makine Öğrenmesi Analizleri	69





ŞEKİL LİSTESİ

Sayfa

Şekil 4.1. Mail Server Trafik Örneği (Zabbix Zirvesi, 2021).....	41
Şekil 4.2. Trend, Season ve Resid Grafikleri (Zabbix Zirvesi, 2021)	42
Şekil 4.3. Sapma Grafik Örneği (Zabbix Zirvesi, 2021).....	43
Şekil 4.4. Periyot ve Season Gösterimi (Zabbix Zirvesi, 2021)	44
Şekil 4.5. Baseline Oluşturma (Zabbix Zirvesi, 2021)	44
Şekil 4.6. Zabbix Sistem Mimarisi (Evonsys, n.d.)	50
Şekil 4.7. Zabbix ile ApplicationServer Sunucu İzlemesi (Evonsys, n.d.).....	51
Şekil 5.1. Alarm Bölümü Autoregistration Ayarlanması	53
Şekil 5.2. Linux Auto Register Tanımı.....	54
Şekil 5.3. Auto Register Şartlarının Belirlenmesi.....	54
Şekil 5.4. Şartlar Oluşturduğunda Alınacak Aksiyonlar	54
Şekil 5.5. Sunucu Üzerinde Tanımlı Örnek Alarmlar.....	56
Şekil 5.6. Zabbix Şablon Kullanımı.....	58
Şekil 5.7. Şablon İçerisinde Bulunan Tanımlar	58
Şekil 5.8. Alarm Tanımları	58
Şekil 5.9 Örnek Alarmlar	59
Şekil 5.10. Alarm Gönderim Tanımları	60
Şekil 5.11. Alarm Konfigürasyonları.....	60
Şekil 5.12. İzlenen Verilerin Gösterimi	60
Şekil 5.13. CPU Yüksek Kullanım Alarmları.....	61
Şekil 5.14. CPU Yüksek Kullanım Grafiği.....	62
Şekil 5.15. Memory Yüksek Kullanım Alarmları.....	62
Şekil 5.16. Memory Yüksek Kullanım Grafiği.....	62
Şekil 5.17. Interface Düşük Kullanım Alarmları	63
Şekil 5.18. Interface Sent Düşük Kullanım Grafiği.....	63
Şekil 5.19. Interface Receive Düşük Kullanım Grafiği	63
Şekil 5.20. CPU Yüksek Kullanım Alarmları.....	64
Şekil 5.21. CPU Yüksek Kullanım Grafiği.....	64
Şekil 5.22. CPU Yüksek Kullanım Alarmları.....	64
Şekil 5.23. CPU Yüksek Kullanım Grafiği.....	65
Şekil 5.24. Disk Kullanımı Artış Alarmları	65
Şekil 5.25. Disk Kullanımı Artış Grafiği	65
Şekil 5.26. Tahminleme Alarmları	66
Şekil 5.27. Cpu ve Memory Kullanımının Gelecekteki Tahminlemesi.....	66
Şekil 5.28. CPU Kullanımının Gelecekteki Tahminlemesi ile Karşılaştırılması.....	66



VERİ MERKEZLERİ İÇİN MAKİNE ÖĞRENMESİ TEMELLİ İZLEME SİSTEMİ TASARIMI

ÖZET

Bugün, hızla büyüyen ve giderek daha karmaşık hale gelen veri merkezleri, modern işletmelerin ve teknolojik altyapıların merkezinde yer almaktadır. Bu büyüme ve karmaşıklık, sistemlerin yönetimi ve operasyonel verimliliklerinin sağlanması için yeni zorluklar ortaya çıkarmaktadır. Bu bağlamda, veri merkezlerinde yer alan sistemlerin sürekli izlenmesi, performanslarının takip edilmesi ve üretilen büyük miktardaki verinin anlamlandırılması kritik bir gereklilik haline gelmiştir. Geleneksel izleme yaklaşımları, belirli sınırlamaları beraberinde getirirken, bu yöntemler, büyük veri hacimleri ve karmaşık altyapılar karşısında yetersiz kalabilmektedir. Bu nedenle, izleme süreçlerinde yenilikçi teknolojiler kullanılarak daha akıllı ve etkin çözümler geliştirilmesi büyük bir önem taşımaktadır. Makine öğrenmesi tabanlı izleme sistemleri, geleneksel yöntemlere kıyasla çok daha hızlı ve doğru bir şekilde verileri analiz edebilmekte, öngörüler yapılarak potansiyel sorunları henüz ortaya çıkmadan tespit edebilmektedir. Bu sistemler, otomasyon sayesinde insan müdahalesine olan gereksinimi azaltarak operasyonel maliyetleri düşürmekte ve insan hatalarından kaynaklanan riskleri minimize etmektedir. Bu da proaktif bakım ve müdahale imkânlarını artırarak sistemlerin sürekliliğini ve performansını en üst düzeye çıkarmaktadır. Erişilebilirlik ve kontrol süreçlerinin kolaylaştırılması, yönetsel yükü azaltırken, izleme sistemlerinin daha kapsamlı ve derinlemesine analiz yeteneklerine sahip olması, karar alma süreçlerinde büyük katkılar sağlamaktadır. Sonuç olarak, veri merkezlerinin izlenmesinde daha analiz odaklı bir yapı oluşturulması, sadece mevcut sistem performansının iyileştirilmesini değil, aynı zamanda gelecek stratejilerinin planlanmasında da önemli bir rol oynamaktadır.

Bu çalışma da Zabbix açık kaynak izleme aracı ve makine öğrenmesi kullanılarak sistemler tasarlanmış ve uygulanmıştır. Her iki izleme yöntemine ait veri toplama yapıları oluşturulmuş, bir sunucu üzerinde Zabbix agent yöntemi ile CPU, memory, disk ve I/O trafiğinin izlenmesi sağlanmıştır. Bu veriler üzerinde hem geleneksel izleme hem de makine öğrenmesi yöntemleri kullanılarak analizler sağlanmış geçmişe ve geleceğe yönelik tahminlemelerde bulunulmuştur. Bu çıktılar sonucunda yansıyan alarmlar ve tutulan verilere göre grafiksel olarak karşılaştırılmış her iki sistemin de avantajları ve dezavantajları ölçümlenmiştir. Geleneksel izlemede sabit formüller yer alırken ve eşik değerler ile sistem izlenirken; makine öğrenmesi ile izleme de geleceğe yönelik tahminleme, baseline oluşturma, doluluk tahminleri gibi analiz edilen, verilerin belirli filtre ve işlemlerden geçirilmesi ile modellemeler yapılmakta ve bu çıktılar sistemin efektif ve proaktif olarak izlenmesine imkân sunmaktadır. Geleneksel bir CPU kullanım alarmı gelirken, geçmişe yönelik baseline değerlendirmeleri ile alarmın yansımadığı olağan olduğu testlerde görülmüştür. Ayrıca geleneksel izleme sistemlerinde genellikle bir eşik değerin aşılması değerlendirilirken makine öğrenmesi ile izlemelerde düşüşlerinde false-positive alarmlar oluşmadan değerlendirilebilmesine imkan vermektedir. Bu da sistemlerde anlık dalgalanmaları

sadece yukarı yönlü değil aşağı yönlüde takip edebilmeyi ve nokta atışı sorunları tespit edebilmeyi sağlamaktadır. Bunun yanı sıra özellikle disk doluluk ve aşım tespitlerinde tahminlemelerin yapılması ve disk, storage gibi yapıların ne zaman dolabileceğinin izlenmesi de kritik öneme sahiptir. Bu bilgi sadece bir alarm olarak kalmamakta ve operasyon planlamalarına destek sağlamakta ve olası kesintilerin önüne geçilmesine imkân tanımaktadır. Veri merkezlerindeki adam-gün işlerinin planlanmasına destek olmaktadır. Çalışma da çıktılar değerlendirildiğinde reaktif yaklaşımlar yerine proaktif yaklaşımların ne gibi faydalar sağlayacağı tasarlanmış olduğumuz sistem üzerinde analiz edilerek ve izlemeleri sağlanarak örneklendirilmiş ve detayları incelenmiştir. Geleceğe bakış olarak makine öğrenmesi sistemlerinde tepesinde bulunan yapay zeka sistemlerinde ne kadar önemli olduğu literatür araştırmalarında görülmüştür. Yeni nesil ve ilerleyen zamanlarda reaktif yapıların yerini makine öğrenmesi ve bir sonraki adımda ise yapay zekanın alabileceği bu çalışma da gözlemlenmiştir.



MACHINE LEARNING BASED MONITORING SYSTEM DESIGN FOR DATA CENTERS

SUMMARY

With today's rapidly developing technological infrastructure, data centers play a critical role. These centers form the backbone of modern businesses and evolve into a constantly growing and complex structure. With the acceleration of digital transformation, the importance of data centers is increasing day by day, which brings with it new management challenges. Continuous monitoring of system performance, analysis and interpretation of large data volumes have become a necessity. While traditional monitoring approaches are inadequate in the face of increasing data volumes and complex infrastructures, the use of innovative technologies has become inevitable. In recent years in particular, the size and complexity of data centers has increased exponentially, making manual monitoring and management processes impractical. Modern data centers host hundreds or even thousands of servers, network devices and storage units, each of which constitutes complex systems that must be constantly monitored and managed. Considering the dependence of businesses on digital infrastructure today, it is clear that even a one-hour system outage can lead to serious financial losses. Interruptions in data center operations can directly cause loss of revenue, customer dissatisfaction, and reputation loss. Monitoring systems, which are critical for business continuity, detect potential problems before they occur and allow the necessary precautions to be taken. Machine learning methods include supervised and unsupervised learning approaches. While supervised learning trains models using labeled data, unsupervised learning aims to discover hidden structures in data by working with unlabeled data. In supervised learning, the model learns the correct outputs together with the input data and makes predictions that can be applied to new data. For example, supervised learning can be used to predict the future performance of a server using past performance data. Unsupervised learning, on the other hand, is used to discover the natural structure of the data. In this method, hidden patterns and relationships in the dataset are revealed. For example, anomaly detection in large data sets such as server logs can be performed using unsupervised learning techniques. Deep learning is an approach that can make more sophisticated predictions and analyses by creating deeper and multi-layered structures in complex data sets. Deep learning can make higher accuracy predictions in large data sets using neural networks. This method achieves successful results especially in complex structured data such as image and audio data. In data center monitoring systems, more complex and variable system behaviors can be analyzed using deep learning techniques and future abnormal situations can be predicted more accurately.

Threshold-based approaches, which form the basis of traditional monitoring systems, are insufficient in the face of today's dynamic workloads. Variable resource usage of modern applications, microservice architectures and cloud infrastructures require more sophisticated monitoring and analysis methods. In this context, machine learning-based approaches offer smarter and adaptive monitoring solutions by learning the

behavioral patterns of systems. In this study, both traditional and machine learning-based monitoring systems were designed and implemented using the Zabbix open source monitoring tool. The development of data center monitoring systems has followed a journey starting from simple ping checks to today's complex artificial intelligence-supported systems. During this evolution process, monitoring systems have transformed from tools that only check the status of systems to proactive management and decision support systems. The main reasons for preferring the Zabbix platform are its large user community, rich plugin ecosystem and easy integration with different systems. In the tests conducted, server behaviors, especially under high load, were examined and the performance differences between traditional and machine learning-based monitoring systems were analyzed in detail. Test scenarios were designed to simulate real-world conditions, and different load levels and usage scenarios were evaluated. Various loads were applied during the testing process and the behavior of the system under different load conditions was examined.

The study demonstrates the critical importance of machine learning-based estimations, especially in disk occupancy and overflow detection. In large data centers, management of storage systems and capacity planning are of critical importance. While traditional approaches generally use simple linear projection methods to estimate capacity, machine learning-based systems can provide more accurate predictions by taking into account usage patterns, seasonal changes, and trend analysis. Thanks to these predictions, occupancy processes of structures such as disks and storage can be predicted in advance, which supports operational planning and allows possible system outages to be prevented. Thanks to the learning ability of the system, disk usage characteristics of different applications can be analyzed, which allows more accurate capacity planning. These predictions also provide great benefits in planning man-day work in data centers. For example, predicting that a storage system will be full within a certain period of time gives IT teams the opportunity to prepare for the necessary capacity increase. This proactive approach minimizes emergency interventions and allows for planned and controlled system updates. In the analyses conducted, it has been observed that machine learning-based prediction systems can make more accurate capacity estimates than traditional methods. This prevents unnecessary capacity increases and minimizes interruptions due to capacity insufficiency. The functions in the Zabbix platform are powerful tools that support machine learning and forecasting processes. For example, the `baselinewma` function is used to determine the baseline of data sets using moving averages. This method is effective in determining the value considered normal in data sets and allows the detection of abnormal deviations. The `TrendSTL` function parses time series data for seasonal trend analysis. This function helps to evaluate system performance more accurately by analyzing seasonal changes and long-term trends in data sets. The `Forecast` function uses statistical models to forecast future data points. This function is used to predict future resource usage and performance trends. The `TimeLeft` function estimates how long a resource will be sufficient based on current data and trends. These estimates help system administrators make more informed decisions and optimize their operational planning. Performance tests and analyses have shown that machine learning-based systems can provide earlier warnings and reduce false-positive alarm rates compared to traditional approaches. These improvements increase operational efficiency while significantly reducing the workload of IT teams. As a result of performance tests and analyses, it has been observed that machine learning-based systems, whose role in data center operations is clearly demonstrated, can adapt to changing system behaviors with their adaptive capabilities. This adaptive feature

ensures that the system operates effectively even in the face of dynamic workloads and changing usage patterns. In the tests conducted within the scope of the study, the superiority of machine learning-based systems, especially in the analysis of complex system behaviors, has been clearly demonstrated. For example, complex service interactions in microservice architectures and their effects on system performance are among the situations that are difficult to detect with traditional monitoring methods. Machine learning-based systems can learn such complex interactions and detect abnormal behaviors. In addition, significant advantages are provided in terms of system security. For example, abnormal system behaviors that may be caused by cyber attacks can be quickly detected by machine learning models. In the future, it is expected that such systems will develop further and evolve into fully automatic and self-healing data center management systems. These systems will not only detect potential problems, but will also be able to automatically produce and implement solutions. This development will increase the efficiency of IT operations, minimize human-related errors and maximize system reliability.

The results of the study demonstrate the benefits of proactive approaches instead of reactive approaches with concrete examples. Especially in large-scale data centers, it seems inevitable that automated, intelligent systems will replace manual control and interventions. Literature research emphasizes the future importance of artificial intelligence systems, which are the upper level of machine learning systems. Artificial intelligence systems will not only detect and predict anomalies, but will also have automatic problem-solving and self-healing capacity. In the future, it is predicted that reactive structures will be replaced first by machine learning, and then by artificial intelligence systems. This transition process marks the beginning of a new era in data center management. It is predicted that artificial intelligence and machine learning technologies will play an even more central role in the future of data center operations. Especially with the widespread use of edge computing and IoT devices, the complexity of monitoring systems and the amount of data that needs to be managed will increase. In this case, the importance of intelligent monitoring systems will increase even more and will play a critical role in the sustainability of data center operations. In addition, with the widespread use of the green data center concept, the optimization of energy consumption is also gaining importance. Machine learning-based systems can contribute to reducing the carbon footprint of data centers by optimizing energy consumption. As a result, the analyses conducted and the results obtained within the scope of this study clearly reveal the advantages of using machine learning technologies in data center monitoring systems. It seems inevitable that traditional monitoring approaches will be replaced by more intelligent and proactive systems. In this transition process, it is of great importance for organizations to adapt to new technologies and manage the necessary transformation processes. In future studies, it is recommended that the role of artificial intelligence technologies and machine learning in data center management be examined in more detail and new usage scenarios be developed. Artificial intelligence and machine learning are among the most critical technologies shaping the future of data centers. The advanced analysis and prediction capabilities offered by these technologies enable system administrators to make more conscious and proactive decisions. In this context, the adoption and integration of these innovative technologies is of great importance for the effective management of data centers.



1. GİRİŞ

Günümüzde veri merkezleri, modern işletmelerin ve dijital altyapıların temel taşı haline gelmiştir. Yeni nesil veri merkezleri, veriyi verimli bir şekilde depolamak, işlemek ve yönetmek için dizayn edilmiş, modern, gelişmiş bir yapıya sahip olmalıdır. Küresel dijital ekonominin büyüyen nabızı, veri merkezlerini yalnızca teknik bir altyapı bileşeni olmaktan çıkarıp, stratejik rekabet unsuru haline getirmiştir (Günel, 2014). Bu dijital dönüşümde kritik parametreler bulunmaktadır. Bunlardan bahsetmek gerekirse ilk olarak veri hacmindeki patlamadan dolayı 2023 yılı itibarıyla günlük üretilen veri miktarı 2.5 kentilyon bayta ulaşmıştır ve her 2 yılda bir veri hacminin ikiye katlanması öngörülmektedir (Arslan, 2024). İkinci sırada ise karmaşık ağ yapılarının olması ve daha da büyümesi. Örneğin; Bulut bilişim entegrasyonları, dağıtık mimari sistemleri ve sürekli değişen teknolojik altyapılar yapıyı daha da karmaşık hale getirmekte ve geleneksel izleme yöntemleri proaktif izlemeyi sınırlandırmaktadır. Geleneksel izleme sistemlerinin yaklaşımından bahsedecek olursak, büyük ölçekli ve dinamik veri merkezi ortamlarında giderek yetersiz kalmaktadır. Çünkü bu sistemler; reaktif çözüm mekanizmaları sunmakta, geriye dönük raporlama yapmaktadır. Anlık müdahale imkanını da sınırlandırmaktadır. Buradan yola çıkılarak makine öğrenmesi ile yenilikçi çözümler üretilmeye çalışılmıştır. Araştırmalara göre, makine öğrenmesi teknolojileri veri merkezi operasyonlarında devrim niteliğinde avantajlar sağlamaktadır (Zhang, 2024). Bu verilere bakacak olursak %92 doğruluk oranıyla gerçek zamanlı anomali tespiti yapılabildiği, proaktif sorun öngörüm mekanizmaları ile olaylar örgüsü başlamadan yakalanabilmesi ve sorun tespiti ve bununla da birlikte yapı uygun ise dinamik kaynak optimizasyonu sağlanabilmektedir. Ayrıca yüksek veri merkezi maliyetlerinden dolayı da tahminsel bakım stratejileri de sunmaktadır. Zhang (2024) Artan veri hacmi, karmaşık ağ yapıları ve kesintisiz hizmet gereksinimleri, veri merkezlerinin etkin bir şekilde izlenmesini ve yönetilmesini kritik hale getirmiştir. Geleneksel izleme sistemleri, büyük ölçekli ve dinamik veri merkezi ortamlarında ortaya çıkan sorunları öngörmede ve çözmede yetersiz kalmaktadır. Bu bağlamda, makine öğrenmesi teknolojileri, veri merkezi operasyonlarının optimize edilmesi ve potansiyel sorunların proaktif olarak tespit edilmesi için yenilikçi çözümler

sunmaktadır. Veri merkezlerinin karmaşıklığı ve kritik önemi göz önüne alındığında, sistem performansının sürekli olarak izlenmesi, analiz edilmesi ve iyileştirilmesi gerekmektedir. Geleneksel reaktif yaklaşımlar yerine, makine öğrenmesi tabanlı proaktif izleme sistemleri, potansiyel sorunları henüz ortaya çıkmadan tespit edebilmekte ve sistem yöneticilerine erken uyarı mekanizmaları sunabilmektedir. Bu yaklaşım, sistem kesintilerinin minimuma indirilmesini, kaynak kullanımının optimize edilmesini ve hizmet kalitesinin artırılmasını sağlamaktadır. Örneğin; DCN üzerindeki veri ve işlemlerde ML destekli sistemlerin somut kazanımlarını ortaya koymaktadır (Dong ve ark, 2021). Makine öğrenmesi tabanlı proaktif izleme sistemlerinin stratejik avantajlarından bahsedecek olursak en başta erken uyarı mekanizmalarının kurulabilmesi çok büyük önem arz etmekte ve asıl farkı ortaya çıkarmaktadır. Bu da potansiyel sorunların henüz oluşmadan tespiti, anlık müdahale yeteneği ve kesintisiz hizmet sürekliliğine imkân sunmaktadır. İkinci olarak kaynak optimizasyonlarının yönetilmesinde de rol alabilmesidir. Dinamik kaynak tahsisi ile yapıların daha verimli hale gelmesi, maliyet etkin olacak şekilde altyapı yönetimi ve bunun yanı sıra ölçeklenebilir performans yönetimini de sağlamasıdır. Gelecek perspektifinden bakacak olursak, yapay zekâ destekli altyapı izleme sistemlerinden şu şekilde bahsedilmektedir: otonom veri merkezlerine dönüşümün olması, minimal insan müdahalesi ile sistem yönetimi ve sürdürülebilir bilişim stratejilerinin daha da önem kazanması şeklindedir (Olaoye ve Potter, 2024).

1.1. Motivasyon

Veri merkezlerinde makine öğrenmesi tabanlı proaktif izleme sistemlerinin uygulanması, geleneksel reaktif izleme yaklaşımlarına göre sistem performansını optimize eder, potansiyel anomalileri daha erken tespit ederek kesinti sürelerini minimize eder, kaynak kullanımını daha verimli hale getirir ve genel operasyonel maliyetleri azaltır. Bu izleme sistemleri, Konteyner veya dinamik ortamlarda ölçeklenebilir ve sürdürülebilir çözümler sunarak, veri merkezi yöneticilerine gerçek zamanlı karar destek mekanizmaları sağlayarak operasyonel verimliliği artırır. Ayrıca, makine öğrenmesi algoritmaları sayesinde, veri merkezi altyapısındaki uzun vadeli performans eğilimlerini tahmin ederek, proaktif bakım stratejileri geliştirilmesine olanak tanır ve sistemlerin güvenilirliğini artırır.

1.2. Problem Tanımı

Geleneksel izleme yöntemleri, veri merkezi ortamlarının performans ve güvenilirliğini sağlamak için genellikle sabit kurallar ve önceden tanımlanmış eşik değerlerine dayanır. Ancak, bu yöntemler dinamik ve karmaşık yapıya sahip modern veri merkezi altyapılarında yetersiz kalmakta, sistem davranışındaki anlık değişimleri algılamakta başarısız olmaktadır. Bu durum, yanlış pozitif (false positive) alarmların artmasına, önemsiz olaylar için gereksiz uyarılar üretilmesine veya kritik sistem hatalarında hiç alarm verilmemesine neden olmaktadır. Böylece, sistem erişilebilirliği ve hizmet sürekliliği olumsuz etkilenmekte, beklenmeyen kesintiler veya performans düşüşleri yaşanabilmektedir. Bunun yanı sıra, manuel müdahale gerektiren bu yöntemler daha fazla insan gücü ve operasyonel efor gerektirir, bu da zaman ve maliyet açısından verimliliği düşürürken, insan hatası riskini de artırmaktadır. Dolayısıyla, geleneksel izleme yaklaşımları, modern veri merkezi operasyonlarının artan ihtiyaçlarına yanıt vermekte yetersiz kalmakta ve daha akıllı, öngörüye dayalı çözümlere olan ihtiyacı ortaya koymaktadır.

1.3. Tezin Kapsamı

Bu tez çalışması, veri merkezlerinde makine öğrenmesi tabanlı izleme sistemi tasarımı odaklanmaktadır. Çalışma kapsamında, gerçek zamanlı veri toplama ve analiz mekanizmalarının geliştirilmesi, ölçeklenebilir sistem mimarisinin oluşturulması ve rol tabanlı izleme mekanizmalarının implementasyonu ele alınmaktadır. Ayrıca, tahmine dayalı izleme teknikleri, sistem anomalilerinin erken tespiti için fonksiyonların tasarlanması ve performans metriklerinin analizi gibi makine öğrenmesi temelli çözümler sunulmaktadır.

Tez kapsamında, IT izleme sistemlerinin implementasyonu ve kaynak kullanımının optimizasyonu konuları incelenmektedir. Bununla birlikte, veri merkezi donanım altyapısının fiziksel tasarımı, ağ topolojisinin planlanması, spesifik bulut servis sağlayıcılarına özel implementasyonlar gibi konular tezin kapsamı dışında tutulmuştur. Çalışmanın temel amacı, geleneksel reaktif izleme ve proaktif otomatize edilmiş bir izleme sistemi tasarlanarak veri merkezi operasyonlarının verimliliğini ve güvenilirliğini değerlendirmektir.

1.4. Tezin Amacı

Bu tez çalışmasının temel amacı, veri merkezlerinde makine öğrenmesi teknikleri kullanılarak proaktif ve akıllı bir izleme sistemi geliştirmektir. Geleneksel izleme sistemlerinin reaktif yapısı yerine, potansiyel sorunları önceden tespit edebilen, sistem performansını optimize eden ve kesinti sürelerini minimize eden yenilikçi bir çözüm sunulması hedeflenmektedir. Bu doğrultuda, gerçek zamanlı veri analizi ve tahmine dayalı izleme teknikleri kullanılarak, veri merkezi operasyonlarının güvenilirliğini ve verimliliğini artırmak amaçlanmaktadır.

Çalışmanın diğer önemli amaçlarından bahsedilirse sistem anomalilerini erken tespit eden ve uyarı mekanizmaları oluşturan akıllı izleme sisteminin geliştirilmesi, dinamik sistemlerde kaynakların optimum kullanımını destekleyecek izleme mekanizmalarının tasarlanması, sürekli izleme süreçlerini otomatize eden bir altyapının oluşturulması, makine öğrenmesi modellerinin kullanımıyla sistem performansının iyileştirilmesi ve operasyonel maliyetlerin azaltılması ve veri merkezi yöneticilerine karar destek mekanizmaları sunan entegre bir çözümün sunulması şeklinde sınıflandırabiliriz. Bu amaçlar doğrultusunda geliştirilen sistem, modern veri merkezlerinin karmaşık gereksinimlerini karşılayan, ölçeklenebilir ve sürdürülebilir bir çözüm sunmayı hedeflemektedir.

1.5. Yöntem

Bu çalışmada, geleneksel ve makine öğrenmesi tabanlı izleme sistemlerinin değerlendirilip karşılaştırılabilmesi amacıyla, açık kaynak kodlu bir izleme aracı olan Zabbix tercih edilmiştir. Zabbix platformu üzerinde her iki izleme yaklaşımı için gerekli yapılandırmalar yapılarak, toplanan veriler analiz edilecektir. Geleneksel izleme sistemlerinde, genellikle önceden tanımlanmış kurallar ve sabit eşik değerlerine dayalı yöntemler kullanılmaktadır. Bu yöntemlerde, ağ cihazları, sunucular ve diğer altyapı bileşenleri belirli performans parametreleri üzerinden izlenir; bu parametrelerin tanımlı eşik değerlerini aşıp aşmadığı kontrol edilir. Eşik değerlerin aşılması durumunda sistemler otomatik olarak alarmlar üretir ve ilgili aksiyonların alınması sağlanır.

1.6. Bilime Katkısı

Bu çalışma, geleneksel izleme yöntemleri ile makine öğrenmesi tabanlı izleme sistemlerini Zabbix platformu üzerinde karşılaştırarak izleme teknolojilerine yenilikçi bir bakış açısı kazandırmaktadır. Geleneksel yöntemlerin sınırlılıkları ile makine öğrenmesi yaklaşımlarının sunduğu avantajlar sistematik olarak değerlendirilmiş, makine öğrenmesi tabanlı proaktif izleme sayesinde sistem anomalilerinin erken tespiti ve tahmine dayalı bakım süreçlerinin etkinliği artırılmıştır. Bu sayede veri merkezlerinde kesinti süreleri azaltılarak operasyonel verimlilik ve maliyet optimizasyonu sağlanmıştır. Çalışmada sunulan uygulamalı model, hem akademik araştırmalar hem de endüstriyel uygulamalar için rehber niteliği taşıırken, gelecekte izleme sistemlerinde makine öğrenmesi ve yapay zekâ entegrasyonuna yönelik araştırmalara da temel oluşturacaktır.



2. LİTERATÜR ARAŞTIRMASI

Veri merkezlerinin etkin bir şekilde izlenmesi ve yönetilmesi, günümüzün kritik bilişim altyapılarının sürdürülebilirliği ve kesintisiz işleyişi açısından hayati bir öneme sahiptir. Bu merkezler, sürekli artan veri hacmi ve kullanıcı talepleriyle başa çıkmak için yüksek performans ve güvenilirlik gereksinimlerini sağlamalıdır. Geleneksel izleme sistemleri genellikle belirli kurallar ve eşik değerlerine dayanır. Bu sistemler, ağ trafiği, CPU kullanımı veya bellek tüketimi gibi parametrelerin önceden belirlenen sınırları aşp aşmadığını kontrol eder ve sınırların aşılması durumunda alarm üretir. Ancak bu tür sistemlerde yüksek miktarda yanlış pozitif ve negatif alarm üretme riski vardır. Karmaşık altyapılarda ölçeklenebilirlik sorunları da yaşanabilir. Gerçek zamanlı ve proaktif analizler sunmakta yetersiz kalır. Çünkü reaktif izleme sistemleri genellikle geçmişe dönük analizler sunarak sorunları tespit etmeye çalışmaktadır. Ancak bu yaklaşım, veri merkezlerinde beklenmedik kesintiler yaşanmasına neden olabilecek potansiyel sorunları önceden tahmin edemez (Nardelli ve ark, 2019). Bu durum uzmanları, daha dinamik, esnek ve proaktif çözümler geliştirmeye yöneltmiştir. Bu da makine öğrenmesi tabanlı proaktif izleme sistemlerinin önem kazanmasına nedne olmuştur. Sorunları minimuma indirmek için öngörücü modellerin kullanılması ve dinamik bir izlemenin olması büyük fayda sağlamaktadır. Örneğin sağlık alanında kullanımlarda, Araştırma şirketi Frost & Sullivan, yapay zekanın hasta sonuçlarını %30 ila %40 oranında iyileştirme ve tedavi maliyetlerini %50'ye kadar azaltma potansiyeline sahip olduğunu tahmin ediyor (Ahuja, 2019).

Makine öğrenmesi (ML) tabanlı sistemler, veri merkezlerinde izleme süreçlerini optimize etmek için önemli bir potansiyel sunmaktadır. Geleneksel yöntemlerin aksine, ML modelleri büyük veri kümelerini analiz edebilir, sistemlerdeki anomalileri gerçek zamanlı olarak tespit edebilir ve proaktif bakım için kullanılabilir. Özellikle veri merkezlerinde üretilen veri miktarının büyüklüğü göz önüne alındığında, bu teknolojiler, veri işleme ve model eğitimi süreçlerini kolaylaştırmakta ve önemini ortaya çıkarmaktadır. Bu bağlamda ML algoritmalarının veri merkezi izleme

süreçlerinde doğruluk oranlarına katkı sağlayarak artırdığı ve %90'ın üzerinde başarı sağladığı görülmüştür (Nardelli ve ark, 2019; Farbiz ve ark, 2020).

Veri merkezlerinde kaynak yönetimi, altyapının performansını optimize etmek için kritik bir gerekliliktir. Geleneksel sistemlerde, kaynak tahsisi genellikle statik yöntemlerle izlenerek aksiyon alınırken, ML destekli izleme sistemleri, dinamik kaynak izlemeleri ve yöntemi ile daha verimli bir yapı sunmaktadır (Everett ve Thomas, 2024). Ayrıca ML modellerinin dinamik tahminlerle enerji tüketimini optimize ettiği ve enerji maliyetlerinde %30'a varan tasarruf sağladığı aktarılmıştır (Langley ve Simon, 1995). Makine öğrenmesi ve yapay zekâ (AI) teknolojileri, otonom veri merkezlerinin de inşasına yönelik önemli adımlar atılmasını sağlamaktadır. Bu sistemler, minimal insan müdahalesi ile kendi kendini yöneten altyapılar oluşturmayı hedeflemektedir (Everett ve Thomas, 2024). Otonom veri merkezleri, sürekli izleme, performans iyileştirme ve tahminsel bakım stratejileri ile yüksek erişilebilirlik sağlamaktadır. Bunun yanı sıra, Langley ve Simon (1995) çalışmasında, yapay zekâ destekli sistemlerin daha sürdürülebilir ve çevre dostu bilişim altyapıları için temel oluşturduğunu ortaya koymaktadır. Makine öğrenmesi tabanlı izleme sistemleri, tahminsel analizlerle veri merkezlerinin risk yönetiminde önemli bir rol oynamaktadır. Bu sistemler, geçmiş verilerden öğrenerek gelecekteki sorunları tahmin edebilmekte ve önleyici müdahaleler için sistem yöneticilerine destek sunmaktadır. Bu bağlamda tahminsel modellerin veri merkezi yöneticilerine hızlı karar alma süreçlerinde %45 daha fazla etkinlik sağladığını belirtilmiştir (Farbiz ve ark, 2020). Makine öğrenmesi algoritmaları, anomali tespiti ve tahmine dayalı bakım süreçlerinde devrim yaratmaktadır. Tahmine dayalı algoritmaların, veri merkezi altyapılarında maliyetten tasarruf sağlandığını ve sistem kesintilerini azalttığı gösterilmiştir (Saini ve ark, 2024). Ayrıca yapay zekâ destekli izleme sistemlerinin gerçek zamanlı analiz yetenekleriyle anomali tespitinde doğruluk oranında %95'in üzerine çıktığı görülmüştür (Shabbir ve ark, 2024). Trend verilerine dayalı izleme de sistem performansını değerlendirmek ve gelecekteki ihtiyaçları tahmin etmek için kullanılan bir tekniktir. Zaman serisi analiz tekniklerinin, trend verilerinden gelecekteki performans eğilimlerini tahmin etmesi ve buna göre sistemdeki sorunları önceden tespiti ve ilgili birimlere bildirilmesi önem arz etmektedir (Akgül, 1994). Özellikle ARIMA ve LSTM gibi algoritmalar, zaman serisi tahminlerinde yüksek doğruluk sağlamaktadır. Bulut ve dağıtık sistemlerde izleme, yüksek ölçeklenebilirlik ve düşük gecikme gerektirir. Büyük ölçekli veri merkezi

ağlarında makine öğrenmesi temelli arıza tespit algoritmalarının, reaktif yöntemlere göre daha hızlı ve etkili olduğu ve otomatize şekilde bir eşik değeri oluşturması bunu desteklemektedir (Abualsauod, 2023). Bunun yanı sıra dağıtık bilişim sistemlerinde de kaynak tahsisi ve performans optimizasyonu için makine öğrenmesi tabanlı yaklaşımlar performansa etki etmekte ve sistem kesintilerini en aza indirmektedir (NR ve ark, 2024).

Bu araştırma sonuçları, veri merkezlerinde makine öğrenmesi tabanlı izleme sistemlerinin hem operasyonel maliyetlerin azaltılmasında hem de sistem performansının iyileştirilmesinde stratejik bir rol oynadığını ortaya koymaktadır. Literatürdeki çalışmaların, gelecekte daha akıllı ve sürdürülebilir veri merkezi altyapılarının geliştirilmesi için önemli bir temel oluşturduğu görülebilir. Gelecekte, bu tür sistemlerin, cihazlar ve uygulamalar arasındaki entegrasyonu sağlayarak daha geniş bir ağ içinde sürdürülebilir izleme çözümleri geliştirilmesine katkı sağlaması öngörülmektedir.

2.1. Veri Merkezi İzleme Sistemleri

Veri merkezlerinin etkin bir şekilde izlenmesi ve yönetilmesi, günümüzün kritik bilişim altyapılarının sürdürülebilirliği açısından büyük önem taşımaktadır. Bu bağlamda, geleneksel izleme yöntemleri yerine; makine öğrenmesi ve yapay zekâ entegrasyonunun izleme sistemlerine katkıları, veri merkezlerinde daha güvenilir, verimli ve maliyet-etkin çözümler sunulmasını sağlamaktadır. Özellikle, yapay zekâ destekli izleme sistemleri, anormalliklerin erken tespiti ve veri analitiği gibi alanlarda büyük avantajlar sunmaktadır. Makine öğrenmesi algoritmaları, izleme sistemlerini daha dinamik hale getirerek, veri merkezlerinde meydana gelen potansiyel aksaklıkların proaktif bir şekilde önüne geçmeyi mümkün kılmaktadır. Bu tür gelişmiş sistemler, sistem yöneticilerine yüksek doğrulukla tahminlerde bulunma yeteneği kazandırarak, sorunları erken aşamalarda tespit etmelerine yardımcı olur (Birje ve ark, 2019). Bunun ile otomasyonun ve bütünleşik izleme teknolojilerinin veri merkezi yönetimindeki önemi giderek daha fazla kabul görmüştür. Bu kabul, veri merkezlerinde gerçek zamanlı ve kontekste duyarlı izleme yaklaşımlarının, insan müdahalesini en aza indirecek şekilde tasarlanması gerektiği fikrini benimsemiştir. Bu yaklaşım, iş gücü verimliliğini artırarak, sistemlerin daha hızlı yanıt vermesini sağlamaktadır. Ayrıca, verimli IT operasyonları yönetimi için izleme sistemlerinin

rolü dikkate alınır, mevcut izleme araçlarının sınırlılıkları öne çıkmakta ve yeni nesil izleme yaklaşımlarının gereksinimlerini ortaya çıkarmaktadır. Özellikle bulut ortamlarında izleme yapmanın zorlukları, sistemlerin ölçeklenebilirliğini ve dinamikliğini koruyarak en verimli hale getirilmesi önerim arz etmektedir (Toka ve ark, 2021).

Veri merkezi izleme sistemlerinin sadece mevcut sorunları çözmekle kalmayıp, aynı zamanda geleceğe yönelik öngörülerde bulunma yeteneğine sahip olması gerektiği de günümüz gerçeğidir. Bu bağlamda, veri toplama ve analiz tekniklerinin gelişmesi, veri merkezlerinin verimliliğini önemli ölçüde artırmaktadır. Gelecekte, bu tür sistemlerin, cihazlar ve uygulamalar arasındaki entegrasyonu sağlayarak daha geniş bir ağ içinde sürdürülebilir izleme çözümleri geliştirilmesine katkı sağlaması beklenmektedir.

2.2. Makine Öğrenmesi ve Veri Merkezi Yönetimi

Makine öğrenmesi, büyük ölçekli veri merkezlerindeki karmaşık ilişkileri analiz etmek ve tahmine dayalı yönetim çözümleri sunmak için önemli bir araç haline gelmiştir. Andreolini (2012) çalışmasında, büyük bilgi sistemlerinin gerçek zamanlı izlenmesi için ölçeklenebilir bir mimari önerirken, makine öğrenmesi tekniklerinin anomali tespiti ve performans tahmini gibi kritik konularda etkin bir şekilde kullanılmasını vurgulamaktadır. Bu bağlamda, veri merkezlerinin operasyonel verimliliğini artırmak ve maliyetleri optimize etmek için predictive monitoring (öngörücü izleme) tekniklerinin önemi de artmaktadır. Özellikle Eyþórsson (2017) makalesinde, makine öğrenmesi tabanlı izleme sistemlerinin IT altyapılarındaki performans ve güvenilirliği artırmadaki rolünü incelemektedir. Bu tür sistemlerin proaktif izleme ve karar destek süreçlerinde önemli katkılar sunulduğu belirtilmiştir. Ayrıca, bulut tabanlı ortamlarda bu tür izleme sistemlerinin ölçeklenebilirliğini sağlayan yeni yaklaşımlar üzerine çalışmalar, bu teknolojilerin daha geniş veri merkezleri ortamlarında etkili bir şekilde uygulanabilmesini mümkün kılmaktadır. Bu alanda yapılan araştırmalar, makine öğrenmesinin sadece sistem hatalarını tespit etmekle kalmayıp, aynı zamanda gelecekteki performans sorunlarını tahmin etme konusunda da büyük bir potansiyel taşıdığını göstermektedir. Bu tür sistemlerin uygulanması, veri merkezlerinin verimli çalışmasını ve olası kesintilerin önlenmesini sağlamak için kritik öneme sahiptir.

2.3. Bulut Sistemlerde İzleme

Bulut sistemlerde izleme, bulut altyapıları, platformları ve uygulamalarının performansını, güvenliğini ve maliyetlerini analiz ederek optimize etmek için kullanılan bir süreçtir. Bu süreç, sistemlerin sorunsuz bir şekilde çalışmasını sağlarken olası aksaklıkların önüne geçmek için kritik bir öneme sahiptir. İzleme, CPU kullanımı, ağ trafiği ve bellek kullanımı gibi metriklerin toplanmasından, uygulama loglarının analiz edilmesine ve güvenlik açıklarının tespitine kadar geniş bir alanı kapsar. Ayrıca, kullanıcı deneyimini artırmak ve uygulama performansını iyileştirmek için gerçek zamanlı veri analizi ve görselleştirme araçları kullanılır. İşletmeler bu verilerden yola çıkarak kaynak kullanımını optimize edebilir, gereksiz maliyetleri azaltabilir ve sorunlara proaktif olarak müdahale edebilir. Bununla birlikte, bulut izleme süreçleri dinamik ortamlar, büyük veri hacimleri ve çoklu sağlayıcılar gibi zorluklar içerir. Kaynakların sürekli olarak ölçeklenmesi ve değişmesi izleme sürecini karmaşık hale getirirken, veri güvenliği ve gizlilik konuları da önemli bir endişe kaynağıdır. Bu bağlamda, işletmeler Amazon CloudWatch, Azure Monitor, Prometheus ve Grafana gibi araçları kullanarak bu zorlukların üstesinden gelebilir. Etkili bir bulut izleme stratejisi, sadece operasyonel kesintilerin önlenmesine değil, aynı zamanda sistem performansının sürekli iyileştirilmesine ve maliyetlerin kontrol altında tutulmasına da katkı sağlar.

2.4. Öngörücü İzleme ve Proaktif Yönetim

IT altyapılarının izlenmesi ve yönetilmesinde öngörücü (predictive) tekniklerin önemi, literatürde sıklıkla vurgulanmaktadır. Mohamed ve Sarr (2024) makalesinde, erken uyarı sistemleri ve sistem hatalarının önlenmesi konularında proaktif bir çerçeve sunmaktadır. Araştırmacılar, makine öğrenmesi tabanlı tahmin modellerinin operasyonel verimliliğe etkilerini incelemektedir (Mohamed ve Sarr, 2024). AI ve öngörücü izleme tekniklerinin IT operasyonlarının iyileştirilmesi ve kesinti sürelerinin azaltılmasındaki rolü azımsanmayacak kadar önemlidir (Alam ve ark, 2023). Özetlemek istenirse, mevcut literatür veri merkezi izleme sistemlerinde makine öğrenmesi tabanlı proaktif çözümlerin önemini vurgulamaktadır. Özellikle gerçek zamanlı anomali tespiti, performans tahmini ve kaynak optimizasyonu konularında, akıllı izleme sistemlerinin kritik rol oynadığı görülmektedir. Bununla birlikte, bulut

ortamlarındaki deęişken ve ölçeklenebilir yapılar için uyarlanabilir izleme mimarilerinin geliştirilmesi de öne çıkan bir araştırma alanıdır.

Literatürü incelediğimizde makine öğrenimi tabanlı izleme sistemleri, geleneksel sistemlerin sınırlamalarını aşarak daha proaktif ve etkili bir yaklaşım sunar. Geleneksel sistemler, olay sonrası analiz ve manuel yapılandırma gerektirirken, makine öğrenimi algoritmaları büyük veri kümelerinden anlamlı bilgiler çıkararak anomali tespiti ve performans optimizasyonu sağlar. Bu sistemler, gerçek zamanlı veri analizi ve otomatik öğrenme yetenekleri sayesinde IT operasyonlarının verimliliğini artırır. Anomali tespiti, denetimli ve denetimsiz öğrenme teknikleriyle gerçekleştirilir. Denetimli öğrenme, önceden etiketlenmiş veri setlerine dayanarak belirli olayları sınıflandırırken, denetimsiz öğrenme, veri içindeki kalıpları ve sapmaları otomatik olarak keşfeder. Zaman serisi analizi ve derin öğrenme gibi teknikler de, karmaşık veri yapılarını analiz ederek daha hassas tahminlerde bulunur. Makine öğrenimi tabanlı izleme sistemlerinin uygulanması, veri toplama, model eğitimi ve sistem entegrasyonu gibi süreçleri içerir. Bu süreçlerde doğru veri seçimi ve model optimizasyonu kritik öneme sahiptir. Ancak, sistemlerin karmaşıklığı ve büyük veri hacimleri, ölçeklenebilirlik ve güvenlik gibi zorluklar yaratır. Bu teknolojilerin benimsenmesi, sistem güvenilirliğini artırırken manuel müdahale ihtiyacını azaltır. Gelecekte, izleme sistemlerinin daha da gelişerek özerk hale gelmesi ve insan müdahalesi olmadan operasyonel kararlar alabilmesi beklenmektedir. Zabbix Açık Kaynak aracı ile geleneksel izleme tanımlarını ve Makine öğrenmesi fonksiyonlarını sisteme tanımlayarak anomali durumlarına nasıl tepki verdiği false positive alarmları egale ettiği olaylar karşılaştırılmıştır.

3. YÖNTEM

3.1. Örneklem

Çalışma da geleneksel ve makine öğrenmesi tabanlı izleme sistemlerinin değerlendirilip, karşılaştırılabilmesi adına Zabbix açık kaynak izleme aracı seçilmiştir. Zabbix üzerinde her iki izleme içinde gerekli tanımlar yapılarak veriler gözlemlenecektir. Geleneksel izleme sistemlerinde, önceden belirlenmiş kurallar ve eşik değerlerine dayalı yöntemler kullanılarak çalışılır. Bu tür sistemlerde, ağ cihazları, sunucular veya diğer altyapı bileşenleri belirli parametreler üzerinden izlenir ve eşik değerlerini aşıp aşmadığı kontrol edilir. Eşik değeri aşıldığında, sistemler alarmlar üretir ve aksiyon alması sağlanır. Dijital dönüşümün hızlandığı günümüz teknoloji ekosisteminde, geleneksel izleme araçları artan karmaşıklık ve hacimde veriyi yönetmekte yetersiz kalmaktadır. Yapay zeka ve makine öğrenimi tabanlı izleme sistemleri, bu kritik boşluğu dolduran ve sistemlerin dijital sınırlarını oluşturan bir çözüm olarak öne çıkmaktadır. Makine öğrenmesi (ML) ile izlemelerde ise geleneksel izleme yöntemlerinde olduğu gibi, bu sistemler de veriyi toplar, ancak makine öğrenmesi, bu veriyi daha derinlemesine analiz eder ve insan müdahalesi ve ihtiyacı en aza indirgenerek, verilerdeki örüntüleri öğrenerek tahminler yapabilir. Bu sayede, sistemler anormal durumları daha hassas bir şekilde tespit edebilir ve potansiyel sorunları önceden belirler. Burada popüler olan ve makine öğrenmesi, yapay zeka gibi yöntemleri içeren izleme araçlarından bahsedeceğiz. Gelişen dünyada geleneksel izleme sistemlerinin sınırlarını aşan yapay zeka ve makine öğrenimi tabanlı izleme sistemleri, modern bilişim altyapılarının performans ve güvenilirlik gereksinimlerine yanıt veren devrim niteliğinde bir yaklaşım sunmaktadır. Bu sistemler, statik kurallar ve sabit eşik değerlerinin ötesine geçerek, sistemlerin dinamik ve karmaşık davranışlarını anlamak ve öngörmek için gelişmiş algoritmalar kullanır. Yapay zeka ve makine öğrenimi tabanlı izleme sistemleri, veri toplama ve analiz süreçlerini tamamen dönüştürmüştür. Geleneksel yaklaşımların aksine, bu sistemlerde verilerdeki örüntüleri otomatik olarak öğrenebilir, insan müdahalesi gerektirmeden sistemin kendi kendine adaptif olmasını sağlar. Gerçek zamanlı anomali tespiti yapabilir, potansiyel sorunları henüz etki yaratmadan önce belirleyebilir. Gelecekteki performans ve olası

sorunları tahmin edebilir, proaktif bakım stratejileri geliştirilmesine imkân tanır. Sistemin kendini optimize etmesine olanak sağlayabilir, kaynak kullanımını ve performansı sürekli olarak iyileştirebilir. Bununla birlikte teknolojik altyapıların giderek karmaşıklaştığı günümüzde, bu sistemler, Bulut bilişimde, mikro hizmet mimarilerinde, dağıtık bilişim sistemleri ve Nesnelerin İnterneti (IOT) ağları gibi çeşitli alan ve ölçeklerde uygulanabilmektedir. Makine öğrenimi ve yapay zeka algoritmaları, sistemlerin davranışsal desenlerini anlamak ve proaktif müdahaleler geliştirebilmek için istatistiksel modellemeler, yapay sinir ağları ve derin öğrenme teknikleri gibi çeşitli yöntemler kullanır.

Modern izleme sistemlerinde yapay zeka entegrasyonu, reaktif bir yaklaşımdan proaktif ve öngörücü bir yaklaşıma geçişi temsil etmektedir. Bu dönüşümün getirilerini şu şekilde sayabiliriz. Kesintisiz operasyonel süreklilik, daha düşük bakım maliyetleri, optimize edilmiş kaynak kullanımı, gelişmiş güvenlik ve performans yönetimi ve hızlandırılmış karar alma mekanizmaları gibi kritik avantajları beraberinde getirmektedir. Geleceğin izleme sistemleri, makine öğrenimi ve yapay zeka teknolojilerinin sürekli gelişimi ile daha da akıllanacak, sistemlerin kendini yöneten, uyarlayan ve optimize eden bir yapıya kavuşmasını sağlayacaktır. Yöntem olarak kullanılan sürecin detayları başlıklar halinde detaylandırılmıştır. Sistemin genel işleyişini bahsedilecek olursa veri toplama ve entegrasyon ile başlar ve ilerler.

3.2. Veri Toplama ve Entegrasyon

3.2.1. Veri toplama nedir?

Veri toplama, belirli bir sistemden veya bileşenden düzenli aralıklarla veri elde etme sürecidir. Bu süreç, bir sistemin performansını, durumunu ve verimliliğini izlemek amacıyla yapılır. Veri toplama genellikle sensörler, izleme araçları veya diğer yazılımlar aracılığıyla yapılır ve bu veriler, sistemin sağlıklı bir şekilde çalışıp çalışmadığını anlamak için sürekli olarak toplanır. Örneğin, bir sunucunun işlemci yükü, bellek kullanımı, disk doluluk oranı ve ağ trafiği gibi veriler, gerçek zamanlı olarak izlenebilir. Bu sayede herhangi bir olumsuzluk erkenden fark edilir ve gerekli önlemler alınabilir.

Veri toplama, sistemin her bileşeninden (sunucular, ağ cihazları, depolama birimleri vb.) yapılabilir. Ayrıca, bu veriler genellikle metrikler, loglar ve izler (traces) olarak kategorize edilir. Metrikler, genellikle sayısal veriler (CPU kullanımı, bellek tüketimi,

ağ trafiği vb.) iken, loglar, sistemin etkinliklerini veya hatalarını açıklayan metin tabanlı kayıtlardır. İzler ise uygulamalar arasındaki etkileşimleri izleyen ve süreçlerin zaman içerisindeki akışını gösteren verileridir.

3.2.2. Veri entegrasyonu nedir?

Veri entegrasyonu, farklı kaynaklardan toplanan verilerin birleştirilmesi ve merkezi bir sisteme aktarılması işlemidir. Bu süreç, verilerin birleştirilmesi, dönüştürülmesi ve uyumlu hale getirilmesini içerir. Entegrasyonun amacı, verilerin farklı kaynaklarda dağılmış olması durumunda, tek bir platformda birleştirilip anlamlı hale getirilmesidir. Örneğin, bir ağ izleme aracı, bir depolama cihazından gelen verilerle, sunucu metriklerinden gelen verileri birleştirerek daha kapsamlı bir analiz sunar. Bu tür entegrasyonlar, sistemin tüm bileşenlerinin tek bir çatı altında izlenmesini sağlar.

Veri entegrasyonu, genellikle API'ler, veritabanları ve veri akışları aracılığıyla gerçekleştirilir. Ayrıca, bu süreçte verilerin uyumlu hale getirilmesi için dönüştürme işlemleri (data transformation) yapılır. Verilerin doğru ve etkin bir şekilde entegrasyonu, özellikle makine öğrenmesi ve yapay zeka uygulamaları için önemlidir, çünkü bu veriler doğru analiz ve tahminler yapabilmek için işlenmiş olmalıdır. Veri toplama ve entegrasyon sistemleri, modern izleme çözümleri için kritik bir öneme sahiptir. Bu konudaki çalışmalar genellikle verilerin doğru bir şekilde toplanması ve analiz edilmesi üzerine yoğunlaşmaktadır. Yapay zeka ve makine öğrenmesinin entegrasyonu ile bu sistemler, daha akıllı hale gelir ve sorunları daha hızlı tespit etme yeteneğine sahip olur.

3.2.3. Makine öğrenmesi nedir?

Makine öğrenmesi (ML), bilgisayarların, açıkça programlanmadan veri analizinden öğrenmesini sağlayan bir yapay zeka (AI) dalıdır (Chen, 2024). ML algoritmaları, büyük veri setlerini analiz ederek örüntüleri, ilişkileri ve anormallikleri keşfeder ve bu bilgiyi yeni veriler üzerinde tahminler yapmak için kullanır. İzleme sistemlerinde, makine öğrenmesi, sistemlerin sürekli olarak gözlemlenmesi ve performansının değerlendirilmesinde kritik bir rol oynar. ML, izleme araçlarında genellikle iki temel görev için kullanılır. Birincisi anomali tespiti ile sistem davranışındaki olağan dışı değişiklikleri tespit etmek. İkincisi tahmin ve karar destek yöntemi ile gelecekteki sistem performansını tahmin etmek ve olası sorunları önceden saptamaktır.

3.2.4. Makine öğrenmesi modelleri ve izleme uygulamaları

3.2.4.1. Denetimli öğrenme (supervised learning)

Denetimli öğrenme, geçmiş verilere dayalı olarak model eğitimi yapar. Bu tür modeller, etiketlenmiş veriler üzerinde çalışır, yani her bir giriş verisiyle ilişkili doğru sonuç (etiket) bilinir. İzleme sistemlerinde, denetimli öğrenme genellikle hata ve arıza tahmini yöntemi ile sistemden alınan geçmiş verilerle model, gelecekteki arıza durumlarını tahmin edebilir. Ayrıca Performans değerlendirme yöntemi ile sistem metrikleri kullanılarak, makine öğrenmesi algoritması, gelecekteki performans düşüşlerini tahmin edebilir. Örnek olarak, Karar Ağaçları (Decision Trees) ve Rastgele Ormanlar (Random Forests) verilebilir (Goswami, 2024). Bunların yanı sıra Destek Vektör Makineleride (SVM) anomali tespiti için yaygın olarak kullanılır (Robles-Durazo ve ark, 2018).

3.2.4.2. Denetimsiz öğrenme (unsupervised learning)

Denetimsiz öğrenme, etiketlenmiş veri olmadan çalışır. Bu tür modeller, veri setindeki doğal yapıları ve örüntüleri keşfetmeye çalışır. İzleme sistemlerinde denetimsiz öğrenme anomali tespiti ve veri kümeleme de kullanılır. Anomali tespitinde sistem davranışındaki olağandışı değişiklikler, örneğin ağ trafiği anormallikleri veya CPU kullanımındaki beklenmedik artışlar, denetimsiz öğrenme algoritmaları ile tespit edilebilir. Veri kümeleme de ise sistemden gelen veriler, benzer özelliklere göre gruplanabilir ve bu kümeler üzerinden performans analizi yapılabilir. Örneğin; Gaussian Mixture Models (GMM): Olasılık tabanlı yaklaşım, genellikle verilerdeki gizli yapıları ve anormallikleri tespit etmek için kullanılır (Yu ve ark, 2023).

3.2.4.3. Takviyeli öğrenme (reinforcement learning)

Takviyeli öğrenme, bir ajan (algoritma) ile çevresel bir sistem arasındaki etkileşimi öğrenme sürecidir. Ajan, çevresinden aldığı geri bildirimlere göre davranışlarını optimize eder. İzleme ve yönetim sistemlerinde özellikle dinamik kaynak yönetimi için kaynak kullanımını optimize etmek ve sistemi daha verimli hale getirmek için takviyeli öğrenme kullanılır.

Örneğin, ağ trafiği yönetimi ve yük dengeleme de yapılabilir. Bunun yanı sıra sistem performansı düşmeye başladığında, takviyeli öğrenme algoritmaları, kaynakları yeniden tahsis etmek veya sistem yapılandırmalarını otomatik olarak optimize etmek için kullanılabilir.

3.2.4.4. Derin öğrenme (deep learning)

Derin öğrenme, çok katmanlı yapay sinir ağları kullanarak verilerdeki yüksek seviyeli örüntüleri öğrenmeye çalışır. Derin öğrenme, büyük veri setlerinde ve karmaşık veri yapılarında başarılıdır. İzleme sistemlerinde derin öğrenme; zaman serisi analizleri ile sistem performansı ve olay verilerini analiz ederek, gelecekteki olayları tahmin edebilir (Örneğin, server arızalarını tahmin etmek). Bir diğer yöntem ise görüntü ve video analizi yapılarak fiziksel güvenlik ve video izleme sistemlerinde, derin öğrenme ile olağandışı hareketler veya güvenlik tehditleri tespit edilebilir. Yapay Sinir Ağları (ANNs) da izleme verileri üzerinde yapılan tahminler ve anomali tespiti için kullanılır. Recurrent Neural Networks (RNN) ve Long Short-Term Memory (LSTM) yöntemleri de özellikle zaman serisi verileri ve sistem performansı tahminleri için yaygın olarak kullanılır (Berus ve Yakut, 2024).

3.2.5. Makine öğrenmesinin izleme sistemlerindeki yeri

Makine öğrenmesi, modern izleme sistemlerinin temel bileşenlerinden biri haline gelmiştir. Bu, özellikle büyük veri ve karmaşık altyapıların yönetildiği ortamlarda önemlidir. Makine öğrenmesinin izleme sistemlerinde sağladığı bazı avantajlardan bahsetmek gerekirse öncelikli olarak Erken Uyarı Sistemi değerlendirilebilir. ML algoritmaları, geçmiş verilere dayanarak gelecekteki anormallikleri tahmin edebilir, böylece sistem yöneticilerine erken uyarılar gönderilebilir. Diğer yandan günümüzde de ihtiyaç duyulan otomatik iyileştirmeler de önem arz ediyor. ML, performans iyileştirme ve sorun giderme süreçlerini otomatikleştirerek manuel müdahale ihtiyacını azaltır ve yapıya katkıda bulunur. Veri Kümeleme ve Segmentasyon sağlayarak büyük veri setlerinde anlamlı gruplar ve segmentler oluşturulabilir, bu da daha hedeflenmiş analiz ve raporlama sağlar. Derin öğrenme ve zaman serisi analizleri kullanılarak, daha doğru ve güvenilir tahminler yapılabilir.

3.3. Trend Verilerine Dayalı İzleme Teknikleri

3.3.1. Trend verileri nedir?

Trend verileri, zaman içinde belirli bir sistem veya bileşenin davranışındaki düzenli değişiklikleri ve eğilimleri yansıtan verilerdir. Bu veriler, belirli bir zaman diliminde ölçülen performans göstergelerinin (metrikler) değişimlerini içerir. İzleme sistemlerinde, trend verileri, sistemlerin sağlığını değerlendirmek, potansiyel sorunları tespit etmek ve gelecekteki performansları tahmin etmek için kritik bir rol oynar.

Trend verileri, geçmiş verilerle yapılan karşılaştırmalar ve analizler sonucu ortaya çıkan uzun vadeli değişimleri ifade eder. Bu veriler, genellikle Anomali Tespiti, Performans İzleme, Tahmin ve Proaktif İzlemelerin gerçekleştirebilmesi adına önemlidir ve yoğun olarak kullanılmaktadır.

3.3.2. Trend izleme teknikleri

Trend verilerine dayalı izleme, genellikle zaman serisi verilerini analiz ederek yapılır. Bu teknik, geçmişteki verilerin analiz edilmesiyle gelecekteki performans hakkında tahminler yapmayı mümkün kılar. İzleme sistemlerinde kullanılan temel trend verilerine dayalı tekniklerden alt başlıklar olarak bahsedilecektir.

3.3.2.1. Zaman serisi analizi

Zaman serisi analizi, bir dizi veriyi zaman sırasına göre analiz etme yöntemidir. Trend verilerine dayalı izleme, genellikle bu analizle yapılır. Zaman serisi analizi, belirli bir metrik veya parametrenin (örneğin, CPU kullanımı, ağ trafiği, disk okuma/yazma hızları) zaman içinde nasıl değiştiğini gözler önüne serer. Zaman serisi analizinde kullanılan başlıca Moving Average (Hareketli Ortalama), Exponential Smoothing (Üssel Düzeltme) ve Autoregressive Integrated Moving Average (ARIMA) tekniklerinden bahsedebiliriz. Moving Average (Hareketli Ortalama) zaman serisi verisindeki geçici dalgalanmaları azaltmak ve genel eğilimi görmek için kullanılır. Bu, izleme sistemlerinde daha net bir trend görselleştirmesi sağlar. Exponential Smoothing (Üssel Düzeltme): Daha güncel verilerin daha fazla ağırlık taşıdığı bir zaman serisi analiz tekniğidir. Geçmiş verilere dayalı tahminlerde doğruluğu artırır (Akgül, 1994). Autoregressive Integrated Moving Average (ARIMA): Zaman serisi verileriyle yapılan tahminlerde kullanılan bir başka istatistiksel modeldir. Bu model, belirli bir verinin gelecekteki değerlerini tahmin etmek için geçmiş verileri analiz eder (Thiyagarajan ve ark, 2017).

3.3.2.2. Tahmin modelleri ve kapasite planlama

Gelişmiş trend verilerine dayalı izleme, sistem kaynaklarının gelecekteki ihtiyaçlarını tahmin etmek için kullanılır. Bu teknik, kapasite planlaması ve optimize edilmiş kaynak yönetimi için kritik önem taşır. Linear Regression (Doğrusal Regresyon) modeli ile iki değişken arasındaki ilişkiyi inceleyerek bir trend çizer. Sistem performansını tahmin etmek için elverişlidir. Machine Learning (Makine Öğrenmesi) ile özellikle zaman serisi verileri ve trend analizleri için derin öğrenme algoritmaları

kullanılarak, gelecekteki olaylar veya performans tahminleri yapmada oldukça faydalıdır. Makine öğrenmesi, izleme sistemlerinde özellikle anomali tespiti ve proaktif bakım gibi alanlarda etkili sonuçlar elde edebilir (Chandola ve ark, 2009).

3.3.3. İzleme, veri görselleştirme ve alarm sistemleri

İzleme sistemlerinde alınan verilere göre alarmların oluşturulması sağlanır. Sistemdeki belirli parametreler zaman içinde belirli bir eşik değerine ulaştığında veya aşarsa, bu durum alarm olarak raporlanır. Trendlerin izlenmesiyle, sistemin gelecekteki performansı hakkında önceden bilgi edinilebilir ve olası sorunlar erkenden tespit edilebilir. İzleme verilerinin görselleştirilmesi, izleme sistemlerinin etkinliğini artırır. Grafikler ve görselleştirmeler, kullanıcıların büyük miktarda veriyi hızlı bir şekilde analiz etmelerini sağlar. Bu görselleştirmeler genellikle zaman çizelgeleri ile sistem performansındaki değişimleri zaman içinde göstermek için kullanılır. Yapıda bulunan dağılım grafikleri ile iki parametre arasındaki ilişkiyi göstermek için faydalıdır.

3.4. Veri Merkezi İzlemelerinde Kullanılan Uygulamalar

Veri merkezi izleme uygulamaları, bu altyapıların her yönünü sürekli olarak izler ve denetler, bu sayede olası sorunlar hızla tespit edilip önceden önlenir. Bu başlık altında, veri merkezi izlemelerinde yaygın olarak kullanılan uygulamaların genel yapısını inceleyeceğiz. Veri merkezi izleme uygulamaları, veri merkezlerinin sürdürülebilirliğini ve etkinliğini sağlamak için kritik öneme sahiptir. Bu araçlar, hem sistem performansını optimize eder hem de olası aksaklıkları ve güvenlik tehditlerini minimize eder. Bugün, büyük veri işleme, bulut bilişim ve yapay zeka gibi yeni teknolojilerin yaygınlaşmasıyla birlikte, veri merkezi izleme uygulamalarının rolü daha da önemli hale gelmiştir. İzleme uygulamaları, daha önce hiç olmadığı kadar güçlü ve entegre bir şekilde çalışmaktadır, bu da veri merkezlerinin verimli ve güvenli bir şekilde operasyonlarını sürdürebilmelerini sağlar.

3.4.1. Prometheus ve grafana

Prometheus, açık kaynaklı bir izleme ve alarm oluşturma aracıdır. Sistemlerin performansını ve sağlığını izlemek için zaman serisi verileri toplayan ve saklayan bir araç olarak geniş çapta kullanılır. Prometheus, hizmetlerden düzenli aralıklarla veri çekerek bu verileri bir veritabanında saklar ve sorgu için kullanılabilir hale getirir. Zaman serisi verilerini toplamak ve sorgulamak için tasarlanmıştır. Kubernetes gibi

mikro hizmet mimarilerinde yaygın olarak kullanılır. Grafana ise bu verilerin görselleştirilmesi ve analitik raporlaması için güçlü bir araçtır. Literatürdeki yerine bakılacak olursa yüksek performanslı ve düşük gecikmeli sorgulama yetenekleriyle sistem izleme alanında önemli bir yere sahiptir. Grafana ise kullanıcı dostu arayüzüyle izleme araçlarını daha erişilebilir hale getirir. Verileri zaman içinde toplar ve bu verileri zaman serisi şeklinde saklar. Bu veriler etiketlerle organize edilir (örneğin, CPU kullanımı metriği, sunucu adı, bölge ve örnek gibi etiketlerle saklanabilir). Bunun ile birlikte Prometheus, PromQL adında güçlü bir sorgu diline sahiptir. Bu dil, kullanıcıların zaman serisi verilerini sorgulamalarını, birleştirmelerini ve raporlar veya grafikler oluşturmasını sağlar ve kullanıcıları belirli eşiklere ulaşıldığında uyaracak şekilde yapılandırılabilen yerleşik bir alarm mekanizmasına sahiptir. Verileri "çekme" (pull) modeliyle toplar, bu da yapılandırmayı ve ölçeklemeyi daha kolay hale getirir. Diğer izleme araçları verileri "itme" (push) modeliyle alırken, Prometheus verileri belirli aralıklarla endpointlardan çeker. Prometheus, özellikle Kubernetes gibi modern bulut tabanlı ve mikro hizmet mimarileriyle birlikte, hizmetlerin, altyapının ve uygulamaların izlenmesinde yaygın olarak kullanılır.

3.4.2. Elastic stack (ELK)

Elastic Stack, log yönetimi ve büyük veri analizi için geliştirilmiştir. Elasticsearch, logların aranmasını sağlarken; Logstash verileri işler ve Kibana görselleştirme sağlar. Ayrıca makine öğrenimi destekli anomali tespiti yapılabilir. Logstash ise log verilerini toplar, filtreler ve Elasticsearch'e iletir. Kibana toplanan verilerin görselleştirilmesini sağlar. Literatürdeki yerine bakılacak olursa büyük ölçekli log analizi ve merkezi izleme çözümleri açısından ELK Stack, kapsamlı bir çözüm sunar. Elastic ML ile anomalileri belirleyebilir ve tahmin modelleri oluşturabilir.

3.4.3. Zabbix

Zabbix, açık kaynaklı bir izleme ve alarm yönetim sistemidir. Sistemlerin, ağların, uygulamaların ve hizmetlerin performansını izlemek için geniş bir izleme altyapısına sahip olup, zaman içinde toplanan verileri analiz ederek olası sorunları erken aşamada tespit etmeye yardımcı olur. Zabbix hem küçük ölçekli sistemler için hem de büyük ölçekli altyapılar için uygun bir çözüm sunar. Özellikle öne çıkan noktaları ise gerçek zamanlı izleme ile ağ cihazlarından ve sunuculardan alınan verileri sürekli olarak analiz ederek, sistem performansını gözlemler. Trigger ve şablonlar ile kullanıcıların belirli eşik değerlerine dayalı tetikleyiciler tanımlamasına ve otomatik yanıt

mekanizmaları kurmasına olanak tanır. Grafiksel analiz ile performans eğilimlerini anlamak için veri görselleştirme araçları sağlar (Zabbix Documentation, 2024).

Zabbix, sunuculardaki CPU, bellek ve disk kullanımını izlemek için geniş destek sunar. Switch, router gibi ağ cihazlarının performansı analiz edilir. MySQL, PostgreSQL gibi popüler veritabanlarının performansını değerlendirir. Docker ve Kubernetes ortamlarında çalışan uygulamaların izlenmesi içinde uygundur. Zabbix'in alarm sistemi, önceden tanımlanmış tetikleyiciler ile belirlenen eşik değerleri aşıldığında kullanıcılara bildirim gönderir. Bu bildirimler: E-posta, SMS veya Telegram gibi platformlar üzerinden iletilir. Zabbix, Prometheus, Grafana ve Elasticsearch gibi araçlarla entegre olarak daha güçlü izleme ve görselleştirme sağlar (Prometheus Documentation, 2024). Özellikle Grafana ile entegrasyon, kullanıcıların gelişmiş görselleştirme panelleri oluşturmaya olanak tanır. Zabbix'in veri toplama aşamasındaki başarısını özellikle dağıtık sistemlerdeki kullanımı gittikçe artmaktadır. Zabbix, Prometheus, Grafana ve Elasticsearch gibi araçlarla entegre olarak daha güçlü izleme ve görselleştirme sağlar (Prometheus Documentation, 2024). Bunun yanı sıra içinde barındırdığı makine öğrenmesi fonksiyonları ile proaktif izlemelere, baseline oluşturmaya, akıllı eşik değerlerine ve tahmin yapma gibi yöntemlere imkân sunmaktadır. Tahminsel analiz ile toplanan verilere dayalı olarak gelecekteki eğilimler analiz edilebilir (Li ve ark, 2022).

3.4.4. Dynatrace

Dynatrace, dağıtık sistemlerin ve mikro hizmet tabanlı mimarilerin izlenmesi ve yönetimi için ileri düzey yöntemler kullanan bir APM aracıdır. Platform, yapay zeka destekli analizler, otomatik kök neden analizi ve gerçek zamanlı veri işleme yetenekleriyle öne çıkar. Dynatrace'ın bu yöntemleri, altyapı ve uygulama performans sorunlarını hızlıca tespit ederek çözüm yolları sunar. Özellikle Davis AI adlı özel yapay zeka motoru, karmaşık sorunları anlamlandırmada kilit rol oynar. Dynatrace, gerçek zamanlı izleme ve analizler için tam yığın görünürlük sağlar. Bu, süreçlerin otomatikleştirilmesi ve anormal durumların tahmini gibi işlemleri destekler. Örneğin, bir uygulama performansı sorunu tespit edildiğinde, Dynatrace'ın yapay zeka motoru, kök nedeni otomatik olarak belirler ve çözüm önerileri sunar. İşletme uygulamalarının ve kullanıcı deneyiminin gerçek zamanlı izlenmesi (APM) gibi AWS, Azure, GCP gibi platformlarda çalışan altyapıların etkin yönetiminde rol alır. Kubernetes ve Docker gibi modern container ortamlarında tam görünürlük sağlar ve uygulama

kullanıcılarının davranışlarını ve deneyimlerini izleyerek optimize eder. Dynatrace'ın izleme yöntemleri; root-cause ile sorunları minimum müdahaleyle otomatik olarak tespit edip çözmek için ileri analiz tekniklerini kullanır. Tahminsel izleme ile gelecekte oluşabilecek sorunları tahmin etmek için geçmiş verilere dayalı yapay zeka modelleri uygular. Dynatrace, yapay zeka ve makine öğrenimi yöntemlerini etkin şekilde kullanarak performans yönetimi ve izleme alanında literatürde lider bir araç olarak öne çıkar. Yapay zeka destekli performans yönetimi sayesinde, yüksek ölçekli altyapıların sürekli olarak izlenmesini sağlar ve bu özelliği ile akademik ve endüstriyel çalışmalarda sıkça referans gösterilir. Dynatrace'ın tahminsel analitik yetenekleri, veri merkezlerinin operasyonel etkinliğini artırmak için literatürde önemli bir örnek teşkil eder. Dynatrace'ın kök neden analizi ve sorun giderme süreçlerindeki otomasyon yetenekleri, endüstri 4.0 ve akıllı altyapı yönetimi alanındaki çalışmalarda öne çıkmaktadır.

3.4.5. New relic

New Relic, Uygulama Performansı İzleme (APM) alanında güçlü ve kapsamlı çözümler sunan bir platformdur. Çeşitli altyapılar ve uygulamalar için izleme imkânı sağlayarak sistemlerin genel sağlığını ve performansını optimize eder. Başlıca yöntemlerden bahsedilecek olur ise gerçek zamanlı izleme ile sistem performansını anlık olarak analiz ederek kullanıcı davranışları ve altyapı sorunlarını tespit eder. Bu, yüksek düzeydeki operasyonel verimliliği destekler. Kullanıcıların ihtiyaçlarına göre özelleştirilebilir grafikler ve raporlama imkânı sunar. Dağıtık İzleme ile mikro hizmet tabanlı mimarilerde, isteklerin sistemde nasıl yayıldığını takip eder. Docker ve Kubernetes gibi konteynerleştirme teknolojileri için doğrudan destek sunar. Özellikle karmaşık mikro hizmet altyapılarında sistem performansını optimize eder. Kullanıcı deneyimi üzerinde ayrıntılı analiz yaparak sorunları tespit eder ve kullanıcı memnuniyetini artırır. New Relic, yapay zeka ve makine öğrenimi entegrasyonlarını izleme süreçlerinde etkin bir şekilde kullanır. Bu kapsamda, anomali tespiti, tahminsel analiz ve otomasyon özelliklerini barındırır. New Relic, mikro hizmet tabanlı izleme alanında öncü bir çözüm olarak kabul edilmektedir. Konteyner ortamlarında performans izleme yöntemlerini ele almış ve New Relic'in bu alandaki kapsamlı araçlarını detaylandırmıştır (New Relic Documentation, 2024), platformun özellikle dağıtık sistemlerde izleme ve konteyner altyapılarında hata yönetimi alanındaki katkılarını vurgulamaktadır.

3.4.6. SigNoz ve openTelemetry

SigNoz, modern yazılım sistemleri için açık kaynaklı bir Uygulama Performansı İzleme (APM) çözümüdür. Loglar, metrikler ve izler (traces) gibi farklı veri türlerini birleştirerek sistemlerin performansını ve davranışlarını daha iyi anlamayı sağlar. SigNoz, OpenTelemetry ile sorunsuz bir şekilde entegre olarak bu izlemeleri yapabilir. Bu entegrasyon, uygulamalardan gelen izleme verilerini toplamak için standart API'leri ve SDK'ları kullanır. Toplanan veriler, performans sorunlarını tespit etmek ve kullanıcı deneyimini iyileştirmek için analiz edilir. Kendi sunucularınızda barındırabilir ve özelleştirebilirsiniz. Bu, verilerin tam kontrolünü sağlar. Mikro hizmet tabanlı sistemlerde işlemlerin nasıl yayıldığını anlamak ve performansı optimize etmek için kullanılır. Uygulama performansındaki darboğazları ve anormallikleri tespit eder. Kullanıcıların etkileşimlerini analiz ederek deneyimlerini iyileştirme imkânı sunar. OpenTelemetry entegrasyonu, standartlaştırılmış veri toplama yöntemleriyle SigNoz'u modern APM çözümleri arasında öne çıkarmaktadır. Özelleştirilebilir ve bulut ortamlarına kolayca entegre edilebilir. Açık kaynaklı olduğu için lisanslama maliyeti bulunmamaktadır. OpenTelemetry ise, çeşitli programlama dillerinde çalışabilir ve geniş bir ekosistemle uyumludur. Veri formatını bir standart haline getirerek sistemler arasında tutarlılık sağlar. SigNoz gibi araçlara kolayca veri aktarır ve bu verilerle derinlemesine analiz yapılmasını mümkün kılar. SigNoz ve OpenTelemetry, modern dağıtık sistemlerin izlenmesi için önemli araçlar olarak öne çıkmaktadır: Özellikle modern yazılım sistemlerinde açık kaynaklı ve esnek mimarisiyle hızla benimsenmektedir. SigNoz'u gelişen APM çözümleri arasında önemli bir alternatif olarak göstermektedir. CNCF topluluğu tarafından desteklenen ve benimsenen OpenTelemetry, veri toplamada standart bir çözüm haline gelmiştir. Ayrıca, yapay zeka ve makine öğrenimi tabanlı analizler için önemli bir veri sağlayıcı olarak görülmektedir (Anand, 2023).

3.4.7. Nagios

Geleneksel Nagios, ağ ve sistem izleme çözümleri alanında uzun bir geçmişe sahip, geleneksel bir izleme aracıdır. İlk olarak 1999 yılında geliştirilmiş olan Nagios, sistem ve ağ yöneticilerinin altyapılarının sağlığını ve performansını etkili bir şekilde kontrol etmelerini sağlayan açık kaynaklı bir platformdur. Geleneksel yöntemlere odaklanarak geniş bir kullanıcı tabanına ulaşmıştır. Nagios, altyapının çeşitli katmanlarında izleme ve uyarı sağlamak için şu yöntemleri kullanır: Ağ cihazlarından ve sunuculardan

durum bilgisi toplar. Özellikle ağ protokollerini izlemek ve hata ayıklamak için kullanılır. Kullanıcılar, izleme kapsamını genişletmek için özel eklentiler geliştirebilir veya mevcut olanları kullanabilir. Örneğin, CPU, bellek ve disk kullanımı gibi sistem kaynaklarını izleyen eklentiler mevcuttur. Sunucuların çalışma durumunu izlemek için "ping" veya hizmet tabanlı testler yapar. Kritik durumlarda SMS, e-posta veya özel komutlarla uyarılar gönderir. Fiziksel ve sanal sunucuların performansını ve durumunu kontrol eder. Ağ cihazlarını (yönlendiriciler, anahtarlar) ve ağ protokollerini izler. Veritabanı hizmetleri, web sunucuları gibi kritik bileşenlerin sağlığını değerlendirir. Disk doluluk oranlarını ve depolama sistemlerinin sağlığını kontrol eder. Nagios, izleme sistemleri alanında uzun süredir var olan bir çözüm olarak güvenilirliği ile dikkat çeker:

Literatürde, Nagios genellikle geleneksel izleme sistemlerinin temel bir örneği olarak kabul edilir. Özellikle ağ izleme ve durum kontrolü alanında yaygın olarak kullanılmaktadır (Nagios Documentation, 2024). Açık kaynaklı yapısı sayesinde, kullanıcılar tarafından farklı ihtiyaçlara uyarlanarak çeşitli endüstrilerde benimsenmiştir. Kullanıcıların kendi eklentilerini geliştirebilmelerine olanak tanır. Modüler yapısı, farklı iş ihtiyaçlarına kolayca adapte edilmesini sağlar. Zaman içinde doğrulanmış bir çözüm olarak, kullanıcılar arasında güven kazanmıştır.

3.4.8. Datadog

Datadog, modern bulut tabanlı altyapı ve uygulamalar için bir izleme ve analitik platformudur. 2010 yılında piyasaya sürülen bu çözüm, geliştiriciler ve sistem yöneticilerinin operasyonel verilerini bir araya getirerek kapsamlı bir görünürlük ve analitik sağlar. Metrikler, loglar, izler (traces) gibi farklı veri türlerini tek bir platformda birleştiren Datadog, güçlü entegrasyon seçenekleri ve kolay ölçeklenebilirliği ile öne çıkar. Datadog, bulut ve hibrit altyapılarda şu yöntemleri kullanarak izleme sağlar: Yapay zeka destekli algoritmalarla standart dışı davranışları tespit eder. Bu özellik, performans düşüşlerini veya olası sistem arızalarını önceden tahmin etme olanağı sunar. Uygulama ve sistem loglarını ilişkilendirerek, hataların kök nedenlerini hızlı bir şekilde belirler. Milyonlarca metriği eşzamanlı olarak işler ve bu metrikleri kullanıcılara gerçek zamanlı olarak sunar. Performans analizinde, API çağrıları ve konteyner bazlı uygulamalardan otomatik veri toplar. Bulut Tabanlı İzleme olarak; AWS, Azure ve Google Cloud gibi platformlarla entegre çalışarak altyapı sağlığını kontrol eder. API çağrıları ve yanıt sürelerini analiz ederek hizmet

sürekliliğini optimize eder. Docker ve Kubernetes tabanlı uygulamalarda kapsamlı metrik toplama ve analiz sağlar. Mikro hizmetler ve dağıtık sistemlerde gecikme, hata oranı ve işlem süresi gibi performans metriklerini değerlendirir. Datadog, özellikle modern bulut tabanlı mimarilerde yaygın olarak kullanılan bir izleme platformu olarak tanımlanmaktadır. Büyük ölçekli sistemlerde metrik, log ve izlerin entegre edilerek kolay görselleştirilmesini sağlayan güçlü bir araç olarak ele alınır. Anomali Tespitleri ile makine öğrenimi algoritmaları, geçmiş verilere dayanarak anomali modellerini belirler. Tahminsel Analiz yöntemi ile yapay zeka destekli tahmin modelleri, olası sorunları önceden tespit eder ve aksiyon alınmasını sağlar. Karmaşık veri kümelerini bir araya getirerek, sistem davranışlarının kapsamlı analizini mümkün kılar (Datadog Documentation, 2024)

3.4.9. Arbor edge defense (AED)

Ağ tabanlı Arbor Edge Defense (AED), ağ tabanlı tehdit izleme ve saldırı önleme alanında kullanılan gelişmiş bir güvenlik çözümüdür. AED, DDoS saldırılarına karşı savunma sağlamak ve ağ trafiğini analiz ederek olası tehditleri tespit etmek için geliştirilmiştir. Arbor Networks tarafından geliştirilen bu çözüm, özellikle kritik altyapıların korunmasında etkin bir rol oynar. Güvenlik odaklı bir çözüm olmasının yanı sıra, ağ performansını optimize eden özellikler de sunar. Ağ trafiğini analiz ederek, şüpheli aktiviteleri ve DDoS saldırılarını gerçek zamanlı olarak algılar ve engeller. Ağ performansını artırmak için yük dengeleme ve trafik yönetimi sağlar. Ağ cihazlarından SNMP protokolü aracılığıyla veri toplar ve performans analizlerini detaylandırır. Arbor Threat Intelligence tarafından desteklenerek, en güncel tehdit verileriyle savunma mekanizmalarını güçlendirir. Arbor Edge Defense, güvenlik ve ağ savunması alanında lider bir araç olarak görülmektedir. Literatürde özellikle tehdit algılama ve önleme de DDoS saldırılarının analizi ve engellenmesi konusunda yüksek başarı oranına sahiptir. Ayrıca AED, kritik altyapıların saldırılara karşı dayanıklılığını artıran özellikleriyle öne çıkar. Arbor Edge Defense (AED) saldırıları gerçekleşmeden önce algılama ve önleme kabiliyeti sunar. Hem volumetrik hem de uygulama katmanı saldırılarını tespit eder. Mevcut ağ altyapısıyla kolayca entegre olur. Ayrıca farklı ağ boyutları ve trafiğe göre ölçeklenebilir bir yapıya sahiptir. Yapay zeka ve makine öğrenmesi yöntemi ile de koruma sağlar (Netscout Documentation, 2024)

3.4.10. Amazon cloudwatch

Amazon CloudWatch, Amazon Web Services (AWS) ekosistemine entegre edilmiş bir izleme ve gözlemlleme çözümdür. AWS kaynaklarının, uygulama performansının ve kullanıcı davranışlarının izlenmesini sağlar. Gerçek zamanlı metrikler, loglar ve olay tabanlı alarmlar sunarak bulut tabanlı altyapılar için kapsamlı bir çözüm sunar. AWS servislerinden gerçek zamanlı metrikler toplayarak altyapı sağlığını ve performansını analiz eder. Örneğin, EC2 örneklerinin CPU kullanımı veya S3 veri erişim sayısı izlenebilir. Uygulama loglarını toplayarak sorunları belirleme ve çözme sürecinde yardımcı olur. AWS Lambda ile entegre çalışarak olaylara otomatik müdahale imkanı sunar. Anomali algılama ve trend analizi ile altyapı sorunlarını proaktif bir şekilde çözmeyi hedefler. CloudWatch, anomalileri ve olası sorunları proaktif bir şekilde algılamak için makine öğrenimi tabanlı araçlar sunar. Örneğin: geçmiş verilere dayalı olarak olağandışı metrik değerlerini tespit eder. Sistem yüklerini ve performansını gelecekteki talepler doğrultusunda optimize etmeye yardımcı olur. Amazon CloudWatch, bulut tabanlı sistemlerde entegre bir izleme aracı olarak önemli bir yere sahiptir. Literatürde özellikle şu alanlarda ele alınmaktadır: AWS ekosistemindeki geniş entegre yetenekleriyle kritik bir araç olarak değerlendirilir (Kavis, 2014). AWS hizmetleriyle tamamen entegre çalışır. Kesintisiz ve hızlı bilgi sağlar. Kullanıcıların kendi metriklerini ve loglarını eklemesine olanak tanır. Hem küçük ölçekli projeler hem de büyük kurumsal uygulamalar için uygundur.

3.5. Geleneksel İzleme Sistemleri

Geleneksel izleme sistemleri, genellikle önceden belirlenmiş kurallar ve eşik değerlerine dayalı yöntemler kullanarak çalışır. Bu tür sistemlerde, cihazlar, sunucular ve diğer altyapılar parametreler üzerinden sürekli olarak izlenir ve bu parametrelerin belirli eşik değerlerini aşıp aşmadığı kontrol edilir. Eşik değerler aşıldığında, sistemler genellikle alarmlar üretir, bu da yöneticilerin hızlı bir şekilde aksiyon almasını sağlar. Bu sistemler, genellikle basit ve doğrudan bir yaklaşım sunar, ancak karmaşık altyapılarda sınırlı esneklik sağlayabilir. Geleneksel izleme sistemlerinin temel amacı, mevcut altyapının sağlığını sürekli olarak gözlemleyerek olası aksaklıkları ve sorunları en erken aşamalarda tespit etmektir. Bununla birlikte, bu sistemlerin etkinliği, kuralların doğruluğu ve güncelliği ile doğrudan ilişkilidir; yanlış eşik değerleri veya yetersiz konfigürasyonlar, yanlış alarmlar veya hatalı tespitlere yol açabilir. Bu nedenle, geleneksel izleme sistemleri genellikle basit yapıların ve net süreçlerin

izlendiđi ortamlar için daha uygundur ve zamanla daha gelişmiş algoritmalar ve yapay zeka tabanlı izleme sistemleri ile desteklenmesi gerekebilir. Prometheus, Nagios gibi izleme sistemleri örnek verilebilir.

3.6. Makine Öğrenimi (ML) Tabanlı İzleme

Makine öğrenimi (ML), verileri analiz etme, modelleme ve gelecekteki olası olayları tahmin etme süreçlerinde giderek daha fazla tercih edilen bir yöntem haline gelmiştir. İzleme sistemlerinde, ML algoritmaları özellikle anomali tespiti, trend analizleri ve sistem performansı tahminleri için kullanılır. Geleneksel izleme yöntemlerinde olduğu gibi, bu sistemler de veriyi toplar, ancak ML, bu veriyi daha derinlemesine analiz edebilir ve insan müdahalesine ihtiyaç duymadan, verilerdeki örüntüleri öğrenerek tahminler yapabilir. Bu sayede, sistemler anormal durumları daha hassas bir şekilde tespit edebilir ve potansiyel sorunları önceden belirleyebilir. Makine öğrenimi tabanlı izleme sistemleri, genellikle büyük ve karmaşık veri setlerinde, manuel müdahale gerektirmeyen otomatik öğrenme yetenekleriyle avantaj sağlar. ML algoritmaları, zaman serisi verisi üzerinde yapılan analizler, sınıflandırma algoritmaları ve denetimli/denetimsiz öğrenme yöntemleri kullanılarak etkin bir şekilde uygulanabilir. Bu sayede, izleme sistemleri sadece geçmişteki olaylara değil, gelecekteki olaylara da dair tahminlerde bulunarak daha proaktif bir yaklaşım sergileyebilir. ML tabanlı izleme, ayrıca anomali tespiti, kullanıcı davranış analizi, altyapı kapasite yönetimi gibi alanlarda da büyük faydalar sunar. Sonuç olarak, makine öğrenimi, veri merkezlerinden bulut altyapılarına kadar geniş bir yelpazede modern izleme sistemlerinin temel unsuru haline gelmiştir (Jani, 2023). Dynatrace, ELK, Datadog ve Zabbix kullanım metodlarına göre bu yöntem altında yer almaktadır.

3.7. Yapay Zeka (AI) Tabanlı İzleme

Yapay zeka (AI) tabanlı izleme, modern sistemlerin performansını optimize etmek, sorunları daha hızlı tespit etmek ve daha doğru tahminler yapmak için kullanılan güçlü bir teknolojidir. Geleneksel izleme sistemlerinin ötesine geçerek, AI tabanlı izleme sistemleri, verilerin derinlemesine analiz edilmesini sağlar ve öğrenme algoritmalarını kullanarak otomatik olarak kararlar alır. Bu sayede, sistemler yalnızca mevcut durumu izlemekle kalmaz, aynı zamanda gelecekteki olası sorunları tahmin edebilir ve bu sorunları önceden engelleyebilir. AI tabanlı izleme, özellikle büyük ve karmaşık

altyapılarda kritik öneme sahiptir. Bu tür altyapılarda geleneksel izleme yöntemleri, yüksek hacimli verileri anlamada ve bu verilerden anlamlı sonuçlar çıkararak proaktif çözümler üretmekte yetersiz kalabilir. Ancak AI algoritmaları, bu verileri sürekli analiz ederek anomali tespiti, davranış analizi ve performans tahminleri gibi görevlerde önemli avantajlar sağlar. Ayrıca, AI tabanlı sistemler, zamanla daha doğru hale gelir çünkü her yeni veri parçası, algoritmaların öğrenme süreçlerine dahil olur.

AI tabanlı izleme sistemlerinin en büyük avantajı, sistemin öğrenmeye devam etmesidir. Bu özellik, makinelerin kendi performanslarını sürekli olarak optimize etmelerini ve daha doğru sonuçlar elde etmelerini sağlar. AI tabanlı izleme, özellikle otomasyonun arttığı, dinamik ve yüksek hacimli veri ortamlarında önemli bir rol oynar. AI'nin izleme alanındaki kullanımı, yalnızca anomali tespitinden çok daha fazlasını içerir; bu, kapasite planlaması, kullanıcı davranışı tahminleri, hata öncesi tespitler ve iş süreci optimizasyonlarını da kapsar. Dynatrace, New Relic gibi izleme sistemleri metodlarına göre bu yöntem altında yer almaktadır.

4. SİSTEM TASARIMI VE UYGULAMA

Bu bölümde, Zabbix platformu üzerinde geleneksel izleme yöntemleri ile makine öğrenimi (ML) tabanlı izleme yaklaşımlarını ele alarak detaylı bir analiz sunulacaktır. İzleme sistemleri, IT altyapılarının güvenilirliği ve performansı için kritik öneme sahiptir. Zabbix, geniş entegrasyon seçenekleri, kullanıcı dostu arayüzü ve özelleştirilebilir yapısıyla geleneksel izleme paradigmasını temsil ederken, makine öğrenimi tabanlı yaklaşımlar bu sisteme yenilikçi bir boyut kazandırmaktadır (*Zabbix Documentation, 2024.*). Geleneksel yöntemler, genellikle önceden belirlenmiş kurallara ve eşik değerlerine dayanır. Bu sistemler, olay tetikleme ve alarm yönetimi gibi temel işlevler için etkilidir; ancak, dinamik ve kompleks altyapılarda sınırlamaları bulunur. Öte yandan, makine öğrenimi tabanlı izleme, tahminsel analiz, anomali tespiti ve kendi kendine optimizasyon gibi özelliklerle proaktif bir yaklaşım sunar. Zabbix'in geleneksel yapısını destekleyen temel özellikler arasında gerçek zamanlı izleme, Zabbix Agent, SNMP ve WMI gibi protokollerle veri toplama ve olay temelli tetikleyiciler bulunur. Ancak, makine öğrenimi entegrasyonu ile Zabbix'in trend analizi ve tahmin kapasitesi artırılarak modern ihtiyaçlara uyum sağlanabilir (Rubanova ve ark, 2019). Bu bağlamda, Zabbix üzerinde gerçekleştirilen geleneksel ve makine öğrenimi tabanlı izleme yaklaşımlarının karşılaştırılması, bu iki paradigmanın güçlü ve zayıf yönlerini analiz etmek ve IT operasyonları için daha verimli bir izleme altyapısı tasarlamak amacıyla ele alınacaktır. Zabbix' in bize sağlamış olduğu imkanlar şu şekilde sıralanabilir: Gerçek zamanlı izleme ilk olarak ele alınırsa bir sistemin, cihazın veya uygulamanın anlık olarak durumunu ve performansını takip etmek için kullanılan bir teknolojidir. Bu yaklaşım, kullanıcıların veya yöneticilerin, meydana gelen olaylara hızlı bir şekilde müdahale etmesine olanak tanır. İzleme, genellikle sistemin performansını optimize etmek, hataları önceden tespit etmek ve iş sürekliliğini sağlamak için kullanılır. Gerçek zamanlı izleme, modern iş süreçleri ve teknoloji altyapıları için kritik bir bileşen haline gelmiştir. Özellikle artan veri hacmi ve karmaşıklığı göz önüne alındığında, doğru izleme araçlarının seçimi operasyonel başarıyı doğrudan etkilemektedir. Bundan dolayı da izleme sistemlerinde de sunucu, ağ cihazı, uygulama ve hizmet performansı gerçek

zamanlı olarak takip edilir. Zabbix agent üzerinde pasif ve aktif kontroller ile ve ICMP, JMX gibi protokollerle veri toplanabilmektedir. Bu kontroller Zabbix tarafından da yapılabilir. Ya da Zabbix agent tarafında aktif olarak yapılabilir. Olay tetikleme ve alarm sistemi için önceden belirlenmiş eşikler aşıldığında alarmlar oluşturur. Bu oluşan alarmlar aksiyon bölümündeki tanımlara göre ilgili noktalara; sms, mail gibi yöntemlerle bilgilendirme yapmaktadır. Zabbix üzerinde bulunan Grafik, harita ve raporlama yöntemleri kullanıcı dostu gösterimler sağlar. Zabbix üzerinde Otomatik keşif ile yeni cihaz ve hizmetler izlemeye alınabilir. Servis veya benzeri dinamik yapılar takip edilebilir. Fiziksel ve sanal sunucular, veri merkezleri ve ağ cihazları ve Veri merkezi içerisinde bulunan her türlü cihaz izlemeye ister manuel ister otomatik olarak dahil edilebilir. Uygulamaların sağlığı ve yanıt süreleri gibi APM noktaları belirli tanımlar ile izlenebilir. Özellikle entegre olarak bulunan web izlemeleri bu noktaları izlemede önemli rol oynamaktadır. AWS, Azure gibi bulut platformları için özel çözümler sunar. Hatta Bulut sistemler üzerinde de Zabbix çalışabilir. Güvenlik olaylarının tespiti ve izlenmesi Zabbix tarafından yapılabilir. Ayrıca Zabbix'in kendi desteği ile uygulama üzerinde bulunan zaafiyetler de kurum olarak takip edilmekte ve güncellemeler ile giderilmesi sağlanmaktadır. Kubernetes ve Docker gibi modern platformlarla entegrasyon sağlar. Zabbix, açık kaynaklı ve özelleştirilebilir bir yapıya sahip olduğu için akademik araştırmalarda sıkça kullanılan bir izleme platformudur. Özellikle kaynak dostu olması sayesinde maliyet avantajı sunar. Kullanıcıların kendi özel betiklerini ve yapılandırmalarını ekleyebilmeleri, Zabbix'i modüler sistemler üzerine yapılan çalışmalarda öne çıkarır. Büyük ölçekli ağlar için uygunluğu sayesinde dağıtık sistem izleme ve yönetim literatüründe yer bulur.

4.1. Alarm Sistemi

Zabbix Alarm Sistemi, Zabbix'in temel özelliklerinden biri olup, bir sistemdeki sorunları tespit etmek, izlemek ve bu sorunlara yönelik bildirimler (alarmlar) oluşturmak için kullanılan bir mekanizmadır. Bu sistem, IT altyapısını kesintisiz çalışır durumda tutmak için kritik öneme sahiptir. Alarmlar, Zabbix tarafından tanımlanan tetikleyiciler (triggers) aracılığıyla otomatik olarak oluşturulur.

Zabbix, kural tabanlı alarm yönetimi sağlar. Tetikleyiciler (Triggers) ile izlenen metriklerin belirli eşik değerlerini veya koşullarını ifade eder. Örneğin: "CPU kullanımı %90'ı aştığında alarm oluştur." Tetikleyiciler, kullanıcı tarafından

tanımlanan mantıksal ifadelerle çalışır. Tetikleyicilerin belirttiği koşullar sağlandığında bir problem (problem) oluşturur. Problem çözülürse bu durumu otomatik olarak algılar ve geri dönüş (recovery) bildirimi gönderir. Kullanıcı tanımlı eşiklere göre olaylar tetiklenebilir. Statik veya dinamik eşik değerleri tanımlanabilir. Örneğin, geçmişteki ortalamaya göre anomaliler tespit edilebilir. Sorun tespit edildiğinde sistem yöneticilerini bilgilendirmek için e-posta, SMS, webhook gibi yöntemlerle alarm gönderebilir. Bildirimler, Zabbix'te tanımlanan medya tipleri (media types) aracılığıyla iletilir. Birden fazla koşula bağlı olarak dinamik alarm senaryoları oluşturulabilir. Sorun giderildiğinde otomatik olarak alarmı kapatma sağlanabilir. Belirli bir süre içinde sorun çözülmezse veya belirli koşullar sağlanırsa, Zabbix alarmları üst düzey bir yöneticiye veya farklı bir ekibe yönlendirebilir.

4.2. Diğer Sistemlerle Entegrasyon

Zabbix, açık kaynak bir araç olduğundan farklı sistemlerle entegre çalışarak geniş bir ekosistemi destekler. Örneğin, AWS CloudWatch, Azure Monitor gibi yapılarla API üzerinden veya agentlar ile entegre olabilir ve izlemeye başlar. ITSM özelinde Jira, ServiceNow veya benzeri ITSM araçları ile entegre olabilir hem izleyebilir hemde alarmlar için ticket açabilir. MySQL, PostgreSQL, MSSQL gibi veritabanlarına bağlanabilir ve detaylı izleme, performans izlemesi sağlar. Grafana ile detaylı şekilde entegre olunabilir. Verilerin bir çok şekilde görselleştirmesi sağlanır. REST API ile harici sistemlerden veri çekme veya sistemlere veri gönderme veya komut çalıştırma, curl istekleri gibi esnek şekilde API isteklerine izin verir.

4.3. Makine Öğrenimi

Makine öğrenmesi (ML), veri üzerinden öğrenen ve tahminlerde bulunan bir yapıya sahiptir. Bu teknoloji, verilerden örüntüler çıkararak ve bu örüntüleri kullanarak gelecekteki veriler üzerinde tahmin veya karar verme yeteneği geliştirmeyi amaçlar. Makine öğrenimi (ML) tabanlı izleme, geleneksel yöntemlerin sunduğu kural tabanlı yapıların ötesine geçerek, sistem davranışlarını otomatik olarak öğrenmeye ve değişikliklere uyum sağlamaya odaklanır. Bu yaklaşım, sistemlerin karmaşıklığını anlamada ve ani değişikliklere yanıt vermede önemli bir avantaj sunar. Zabbix platformuna makine öğrenimi yetenekleri eklemek, özellikle anomali tespiti,

tahminsel analiz ve otomatik optimizasyon alanlarında izleme kapasitesini ciddi şekilde artırabilir.

Bu modeller, sistemin normal davranışını öğrenerek bu davranıştan sapmaları yüksek doğrulukla tespit edebilir. Örneğin, bir ağdaki trafik modeli üzerinde çalışarak potansiyel siber saldırıları önceden algılamak veya bir veri merkezinde kaynak kullanımını optimize etmek için proaktif önlemler almak mümkündür. Avantajlarına değinilecek olursa, statik eşik değerlerinden farklı olarak, ML modelleri gerçek zamanlı verilere dayanarak eşikleri dinamik olarak ayarlayabilir ve sistem davranışlarındaki ani değişimlere hızlıca uyum sağlayabilir. Normal davranıştan sapmaları tespit ederek potansiyel sorunları öngörebilir. Örneğin, veri setlerinin normal dışı noktalarını belirleyen denetimsiz öğrenme yöntemleri bu alanda etkilidir (Rubanova ve ark, 2019). Gelecekteki sorunları veya performans düşüşlerini önceden tahmin ederek önleyici eylemler almayı mümkün kılar. Büyük ölçekli sistemlerde bile detaylı analiz yaparak performans ve güvenilirlik sağlar. Dezavantajları ele alınırsa büyük veri setlerinin işlenmesi ve karmaşık algoritmaların çalıştırılması, güçlü işlemciler ve büyük kaynaklara ihtiyaç duyar. Modelin doğru çalışması için büyük ve kaliteli bir veri seti gereklidir. Ayrıca, eğitim süreci zaman alıcı olabilir. Makine öğrenimi algoritmalarının uygulanması, geleneksel izleme sistemlerine göre daha karmaşık bir yapıya ihtiyaç duyulabilir.

Zabbix kendi içerisinde bulundurduğu makine öğrenmesi fonksiyonları ile yeni nesil izlemelere imkan sunmaktadır. Buna örnek olarak tahminsel izleme'yi (Predictive Monitoring) gösterebiliriz. Sistemlerin mevcut ve geçmiş performans verilerini analiz ederek gelecekte meydana gelebilecek sorunları veya anomalileri önceden tahmin etmek için kullanılan bir izleme yaklaşımıdır. Bu yöntem, makine öğrenmesi (ML), istatistiksel analiz ve veri modelleme tekniklerini kullanarak sorunları önceden belirlemeyi ve çözüm için proaktif önlemler almayı amaçlar. Zabbix'in mevcut trend fonksiyonları, zaman serisi verilerinin ileriye dönük analizi için temel sağlar (Zabbix Documentation, 2024). Anomali Tespiti ile bir veri setinde normal davranış veya örüntülerden sapma gösteren olayları, değerleri veya gözlemleri tanımlar. Bu yöntem, normal sınırların dışında kalan verileri belirleyerek olası hataları, saldırıları veya anormal durumları tespit etmek için kullanılır. Makine öğrenimi modelleri ile entegre edilerek normal dışı durumlar algılanabilir. Zabbix üzerinde dahili fonksiyonlar kullanılarak anomali tespitleri yapılabilir. Makine öğrenimi algoritmaları, metrik

eşiklerini ve sistem yapılandırmalarını dinamik olarak optimize edebilir. Harici modeller de kullanılarak ilerlenebilir. Örneğin, TensorFlow, PyTorch veya scikit-learn gibi popüler ML kütüphaneleri ile oluşturulan modeller, Zabbix'in API'leri aracılığıyla sisteme entegre edilebilir. Makine öğrenimi tabanlı bu yenilikler, sadece mevcut sorunların çözümünde değil, aynı zamanda gelecekteki sorunların proaktif bir şekilde önlenmesinde de büyük rol oynar. Bu, özellikle bulut tabanlı sistemler, dağıtık veri merkezleri ve karmaşık IT altyapıları için önemli bir avantaj sunmaktadır. Tablo 4.1 'de temel olarak geleneksel ve ml tabanlı izleme farkları değerlendirilmiştir.

Tablo 4.1. Geleneksel ve ML Tabanlı İzleme Karşılaştırması

Özellik	Geleneksel İzleme	ML Tabanlı İzleme
Tetikleme Mekanizması	Statik eşik değerleri	Dinamik öğrenme ve adaptasyon
Hata Tespiti	Sınırlı	Yüksek
Uygulama Karmaşıklığı	Basit	Karmaşık
Performans Analizi	Mevcut durum	Geleceğe yönelik tahmin
Kaynak Kullanımı	Düşük	Orta-Yüksek

4.4. Genel Tasarım Yaklaşımı

Bu tasarımda esnek, genişletilebilir ve güncel olan bir izleme aracı olan Zabbix'i özellikle seçmiş olduk. Modern veri merkezleri için tasarlanan bir izleme sisteminde, esneklik, genişletilebilirlik ve sürdürülebilirlik en önemli tasarım kriterleri arasında yer alır. Bu bağlamda Zabbix, sadece bir izleme aracı değil, aynı zamanda gelişmiş bir mimari sunarak sistem ihtiyaçlarına uyum sağlayan bir çözüm olarak öne çıkar. Tasarım olarak esnek ve büyümeyi kendi içinde resmi olarak desteklemesi önem arz etmektedir.

4.4.1. Sistemin amaçları ve gereksinimleri

Bu izleme sistemi, veri merkezlerinin artan operasyonel ihtiyaçlarını tespit edebilmek ve karşılamak amacıyla yüksek verimlilik ve izleme hedefleriyle tasarlanmıştır. Gelişmiş bir izleme aracı olan Zabbix proaktif izlemeler ile sorunlar meydana

gelmeden önce algılama ve müdahale imkanı sağlar. Verilerin anlık olarak işlenmesiyle operasyonel gecikmelerin önüne geçilir. Yeni ihtiyaçlara göre kolayca genişletilebilir bir yapı sunar.

4.4.2. Rol tabanlı mimari

Sistemin kullanıcı yetkilendirme ve erişim kontrolü mekanizması, güvenlik ve organizasyonel verimlilik açısından kritik bir öneme sahiptir. Kullanıcı Profilleri ve Rollerini farklı görev ve sorumluluklara sahip kullanıcılar için özelleştirilebilir erişim seviyeleri tanımlanabilir. Kullanıcıların yalnızca görevleriyle ilgili verilere erişmesini sağlayarak veri güvenliğini artırır. Kullanıcı aktiviteleri sistem tarafından sürekli olarak izlenir ve raporlanır, bu da güvenlik ve uyumluluk gereksinimlerini karşılar.

4.4.3. Ölçeklenebilirlik

Bir sistemin, altyapının veya uygulamanın artan iş yüküne veya kullanıcı sayısına uyum sağlayabilme kapasitesini ifade eder. Sistem, işlevselliğini ve performansını koruyarak kaynaklarını genişletebilir veya azaltabilir. Ölçeklenebilirlik, modern IT altyapıları ve yazılım geliştirme süreçlerinde kritik bir tasarım prensibidir. Modern veri merkezlerinde izleme sistemlerinin artan talepleri karşılayabilmesi için ölçeklenebilirlik kritik bir öneme sahiptir. Zabbix, hem yatay hem de dikey genişleme imkanı sunarak esnek bir altyapı sağlar. Yatay olarak yeni sunucular ve cihazlar eklendiğinde sistem kapasitesi artırılabilir. Bu, özellikle büyük veri merkezleri için önemlidir. Dikey büyüme için Artan işlem kapasitesi ihtiyaçlarını karşılamak için mevcut donanım yükseltilebilir. Dağıtık izlemeyi proxyleri ile yaparak birden fazla veri merkezini veya coğrafi olarak dağınık altyapıları tek bir merkezi yapı üzerinden yönetme imkanı sunar.

4.4.4. Gelişmiş entegrasyon ve uzun vadeli uyumluluk

Zabbix, açık API'leri ve geniş entegrasyon yetenekleri sayesinde diğer izleme araçları, yönetim sistemleri ve veri analiz platformları ile uyum içinde çalışabilir. Grafana gibi görselleştirme araçlarıyla kolayca entegre olabilir. ML tabanlı tahminsel analiz ve anormali tespiti gibi gelişmiş özelliklere destek sağlar. Sürekli güncellenen modülleri sayesinde gelişen teknolojiyle uyum içinde çalışır. Bu genel tasarım yaklaşımı, sadece veri merkezleri için değil, karmaşık IT altyapılarında da sürdürülebilir bir izleme çözümü sunmayı hedeflemektedir. Bu yapı, organizasyonların operasyonel

maliyetlerini düşürürken, hizmet kalitesini artıran proaktif bir izleme ve yönetim deneyimi sağlar.

4.5. Makine Öğrenmesi Entegrasyonu

Makine öğrenmesi (ML) entegrasyonu, modern izleme sistemlerini geleneksel yöntemlerin ötesine taşır ve özellikle büyük ölçekli, dinamik altyapılarda kritik bir rol oynar. Zabbix, ML tabanlı yetenekleri kendi işlevselliğine ekleyerek hem dahili hem de harici araçlarla güçlü entegrasyon seçenekleri sunar. Zabbix, makine öğrenimi (ML) tabanlı bazı işlevleri doğrudan destekleyerek daha dinamik ve akıllı izleme süreçlerine olanak tanır. Modern izleme sistemlerini geleneksel yöntemlerin ötesine taşır ve özellikle büyük ölçekli, dinamik altyapılarda kritik bir rol oynar. Trigger fonksiyonları ile ML entegrasyonları yapılmaktadır.

4.5.1. Dahili trend analizi ve tahminsel izleme

Zabbix, sistemdeki geçmiş verileri analiz etmek ve trend tahminleri yapmak için dahili trend fonksiyonlarını kullanır. Bu fonksiyonlar, gelecek kaynak tüketimini tahmin etmek, kapasite planlaması yapmak ve performans sorunlarını öngörmek için önemli bir temel sağlar. Dahili Trend Analizinde toplanan verileri uzun süreli trendler oluşturmak için kullanır ve bu veriler, günlük veya haftalık özetler gibi düşük detaylı ancak etkili bir biçimde saklanır. Detaylı tarihsel veriler yerine yalnızca özetler saklanır, böylece disk alanı kullanımı optimize edilir. Donanım yükseltme gereksinimleri veya kullanıcı trafiğindeki artışlar gibi konular için gelecekteki kaynak ihtiyaçlarını tahmin eder. Örneğin, `trendavg()` ve `trendmin()` fonksiyonları geçmiş verileri baz alarak ortalama veya minimum değerler hakkında bilgi sağlar. Bu değerler, belirli eşiklerin aşılmasını önlemek amacıyla önceden uyarılar oluşturabilir. `trendavg()`, Zabbix'te belirli bir zaman dilimi içinde bir metriğin ortalama değerini hesaplamak için kullanılır. Bu fonksiyon, trend verilerini analiz eder ve uzun süreli gözlemlerde performans eğilimlerini anlamak için idealdir. Örneğin, sunucuların CPU veya bellek kullanımının uzun vadeli ortalamasını görmek veya bir metriğin belirli bir zaman aralığındaki ortalama davranışını tespit ederek eşik değerlerini belirlenmesi sağlanır.

`trendavg(/host/key, time_period)`

`/host/key`: İzlenen metrik (örneğin `system.cpu.util`)

`time_period`: İncelenecek zaman dilimi (örneğin 1h - son 1 saat, 24h - son 24 saat).

`trendmin()`, bir metriğin belirtilen zaman dilimi içerisindeki minimum değerini döndürür. Bu fonksiyon, özellikle sistem performansında ani düşüşleri veya düşük etkinlik periyotlarını analiz etmek için kullanılır. Örneğin sistem kaynaklarının en az hangi seviyede kullanıldığını görmek veya kaynak yetersizliği olabilecek durumları analiz etmek için imkan sunar.

`trendmin(/host/key, time_period)`

item: İzlenen metrik (örneğin `system.cpu.util`)

`time_period`: İncelenecek zaman dilimi (örneğin 1h - son 1 saat, 24h - son 24 saat)

Tahminsel izleme ise, Zabbix'in trend analizinden yararlanarak gelecekteki potansiyel sorunları veya performans sapmalarını önceden belirlemesini sağlayan bir özelliktir. Bu yöntem, geçmiş verilere dayanarak gelecekteki davranışları tahmin eder ve sistem yöneticilerinin erken müdahale etmesine olanak tanır. Zabbix, geçmişteki veri trendlerini analiz ederek, gelecekte bir metriğin ne kadar yükseleceğini veya düşeceğini tahmin eder. Sorunlar oluşmadan önce uyarılar göndererek kesinti sürelerini ve maliyetleri azaltır. Bir depolama biriminin doluluk oranının trendine bakarak, bu birimin 3 hafta içinde tamamen dolacağı tahmin edilebilir ve yeni depolama eklemek için zaman kazanılabilir. Örneğin `forecast` fonksiyonu;

`forecast(/host/key,1h,30m)`

Son saat verilerine göre item değerinin 30 dakikalık tahminini yapar. Zabbix'in dahili trend analizi ve tahminsel izleme özellikleri, sistem yönetiminde proaktif bir yaklaşım sağlar. Bu araçlar, sistem performansını optimize etmek, kaynakları daha iyi yönetmek ve sorunları önceden öngörmek isteyen kuruluşlar için vazgeçilmezdir. Özellikle büyüyen altyapılarda ve dinamik iş yüklerinde, bu özellikler işletmelerin sürdürülebilirlik ve verimlilik hedeflerine ulaşmalarına yardımcı olur.

4.5.2. Baseline oluşturma ve analizi

Baseline, bir sistemin "normal" kabul edilen davranışını ve performansını tanımlayan referans değerler setidir. Bu değerler, sistemin zaman içindeki geçmiş performansı ve belirli metrikler üzerindeki gözlemlerden elde edilir. Zabbix, baseline analizi için yerleşik fonksiyonlar sunar ve bu analiz, anormalliklerin belirlenmesi ve tahminsel bakım yapılması için temel oluşturur. Zabbix, topladığı metrikler üzerinden bir zaman

serisi analizi yaparak baseline oluşturabilir. Bu süreçte, sistemin normal davranışını öğrenmek, gelecekteki performansı öngörmek ve belirli sapmaları hızlı bir şekilde algılamak amacıyla kullanılır. Zabbix'te `baselnewma()` fonksiyonu, bu tür analizlerde sıklıkla kullanılan bir yöntemdir. `baselnewma()`, Zabbix'te ağırlıklı hareketli ortalama (Weighted Moving Average - WMA) yöntemiyle bir baseline oluşturur. Bu fonksiyon metrik verilerinin geçmiş değerlerini analiz ederek, daha güncel verilere daha fazla ağırlık verir. Uzun vadeli eğilimleri hesaba katarak, kısa vadeli dalgalanmaların etkisini azaltır. Örneğin,

`baselnewma(/host/key, 1h, 7d)`

`/host/key` metrik verilerinin, son 1 saatlik eğilimi analiz ederek, Son 7 günlük geçmiş verilerle kıyaslayarak bir baseline belirler. Belirli bir metriğin gerçek zamanlı değeri, baseline ile karşılaştırılır. Örneğin, CPU kullanımının son 1 saatte %80'in üzerine çıkması ve geçmiş 7 günle kıyaslandığında bu değer normalin dışında olması durumunda bir uyarı tetiklenir. Zabbix, zaman içinde oluşan trendleri analiz ederek, gelecekteki davranışları tahmin edebilir. Bu özellik, sistem büyümesi ve kaynak ihtiyaçlarının planlanması için kullanılabilir. Günün belirli saatlerinde veya yılın belirli dönemlerinde tekrar eden metrik davranışları için özel baselines tanımlanabilir. Bu yöntem e-ticaret sitelerinde alışveriş yoğunluğunun artacağı tatil dönemleri için özel eşik değerlerinin belirlenmesinde kullanılabilir. Avantajlarından bahsedilecek olursa geçmiş verilere dayalı analiz, statik eşiklerden daha doğru sonuçlar sağlar. Sistem davranışına özgü eşikler tanımlanabilir. `Baselnewma` ile gelecekte oluşabilecek sorunları önceden öngörebilir. `Baselnewma()` fonksiyonu, sistem davranışını detaylı bir şekilde analiz etmeyi mümkün kılar. Bu analiz, izleme süreçlerini daha akıllı ve etkili hale getirerek operasyonel kararların daha doğru alınmasına olanak tanır.

4.5.3. Dinamik eşik ayarlama ve analizi

Dinamik eşik ayarlama, sistem metriklerinin zaman içindeki normal davranışlarını analiz ederek, bu davranışlara uygun esnek eşik değerleri belirlenmesini sağlar. Geleneksel izleme sistemlerinde statik eşikler, sabit bir limit tanımlayarak uyarı üretir. Ancak bu yöntem trafik yoğunluğunun saatlik değişim gösterdiği ağlarda, sezonsal kullanım artışlarının yaşandığı sistemlerde ve beklenmedik yük dalgalanmalarının olduğu kritik uygulamalarda yetersiz kalabilir. Dinamik eşiklerle, anlık koşullara göre esnek ayarlamalar yapılır ve gereksiz uyarılar azaltılarak doğru kararlar alınır. Trend

analizi ile otomatik eşik belirlemek için Zabbix'in yerleşik fonksiyonları (örneğin, `trendavg()`, `trendmax()`) geçmiş metriklere dayalı analiz yaparak, normal davranış profili çıkarır. Bu profiller öngörülen sistem performansını anlamak ve kritik uyarılar için uygun eşik değerlerini dinamik olarak hesaplamak için kullanılır. Burada dinamik eşikler sayesinde, normalden sapmalar kolayca algılanır ve sistemde sorun olmadan önce uyarılar oluşturulur. Bu da operasyon ekiplerinin gerçek sorunlara daha hızlı odaklanmasına katkı sağlar. Zabbix, anlık metrik değişimlerini ve genel trendleri sürekli gözden geçirerek, eşiklerin zamanla daha hassas hale gelmesini sağlar. Bu süreçte statik parametrelerin güncellenmesine gerek kalmaz, sistem koşullara adapte olur.

4.5.4. Harici makine öğrenimi modelleri ile entegrasyon

Zabbix'in API yetenekleri, harici makine öğrenimi (ML) modelleriyle entegrasyonunu kolaylaştırır. Bu entegrasyon, Zabbix'te toplanan verilerin işlenmesi, analiz edilmesi ve daha karmaşık öngörüler veya anomali tespitleri için ML modellerine aktarılmasını sağlar. Ayrıca, ML modellerinin çıktıları Zabbix'e geri beslenerek, tetikleyiciler ve uyarılar için kullanılabilir. Zabbix API sayesinde veriler, Python gibi bir programlama dili kullanılarak çekilebilir ve bir ML modeli üzerinde işlenebilir. Geçmiş verilerin Zabbix'ten alınır. Model eğitimi için veriler temizlenir ve hazır edilir. Eğitimli modelin sonuçlarını tekrar Zabbix'e entegre ederek operasyonel raporlama ve uyarılar oluşturulması sağlanır.

4.6. Entegrasyon ve Test Süreçleri

Geleneksel izleme tasarımıyla da karşılaştırılarak iki sistemin performans ve etkinlik bakımından değerlendirilmesi yapılacaktır. Çalışmanın temel metodolojisi, veri merkezi altyapısından elde edilen performans verilerinin gerçek zamanlı toplanması, işlenmesi ve analiz edilmesine dayanmaktadır. Makine öğrenmesi algoritmalarıyla desteklenen sistem, anomali tespiti, performans tahmini ve kaynak optimizasyonu gibi kritik işlevler sunarak operasyonel verimliliği artırmayı amaçlamaktadır.

4.7. Veri Toplama ve Entegrasyon

Gerçek zamanlı veri toplama mekanizmaları, sistemlerin durumunu izlemek ve performanslarını değerlendirmek için kritik öneme sahiptir. Bu bağlamda, çeşitli izleme araçları ve sensörler kullanılarak veri merkezi bileşenlerinden (sunucular, ağ

cihazları, depolama birimleri vb.) sürekli veri akışı sağlanacaktır. Toplanan veriler, makine öğrenmesi modellerine uygun hale getirilerek analiz edilecek ve sistemdeki potansiyel sorunlar erken aşamada tespit edilecektir. Zabbix Açık kaynak izleme aracı ile CPU, Memory, Interface ve Disk verileri toplanacak. Toplanan veriler Zabbix aracılığı ile işlenerek anlamlı hale getirilecek ve bu veriler Zabbix veritabanında tutulacak. Veritabanı PostgreSQL'dir.

4.8. Makine Öğrenmesi Modelleri

Çalışmada, sistem anomalilerini tespit etmek için makine öğrenmesi algoritmalarından yararlanılacaktır. Özellikle, denetimli öğrenme yöntemleri kullanılarak, geçmiş veri setlerine dayanarak sistem davranışları modellenip, anomali tespiti yapılacaktır. Bu yöntemle, sistemdeki potansiyel arızalar ve performans düşüşleri, henüz bir arıza gerçekleşmeden önce öngörülebilecektir. Makine öğrenmesi modelleri, zaman serisi verileri üzerinde çalışacak şekilde tasarlanacak ve algoritmalar, eğitim verilerinden öğrenerek doğruluk oranlarını artıracaktır. Kullanılacak spesifik algoritmalar: Baselines WMA, STL (Seasonal-Trend decomposition using LOESS) bu iki algoritma özellikle monitoringteki sorunların tespiti için en ihtiyaç duyulan durumlardır.

4.9. Trend Verilerine Dayalı İzleme Teknikleri

Zabbix'teki trendstl fonksiyonu, zaman serisi verilerindeki trendleri ve mevsimsel bileşenleri analiz etmek için kullanılan bir araçtır. Fonksiyon, özellikle uzun vadeli veri analizinde daha net bir şekilde eğilimleri ve tekrar eden desenleri ayırmak için tasarlanmıştır. Bu fonksiyon, STL (Seasonal and Trend decomposition using Loess) algoritmasını temel alır.

4.9.1. Çalışma prensibi

İzlemenin yapılabilmesi adına veri girdisi Zabbix'teki tarihsel veri veya trend tablosundan alınan zaman serisi olacaktır. Trend ayrıştırma kullanılarak veri içindeki uzun vadeli eğilim (trend) belirlenir. Sonra Mevsimsel ayrıştırma ile verideki dönemsel (mevsimsel) desenler, belirli periyotlarla tekrarlayan bileşenler olarak ayrılır. Rastgele bileşen ile kalan veri, trend ve mevsimsel bileşenlerden arındırıldıktan sonra rastgele hata (residual) olarak bırakılır.

4.9.2. Algoritma detayları

STL algoritması üç ana aşamada çalışır. Belirli bir dönem uzunluğuna göre mevsimsel bileşen belirlenir. LOESS (Locally Estimated Scatterplot Smoothing) kullanılarak veriler yerel regresyonla düzleştirilir. Mevsimsel bileşen çıkarıldıktan sonra kalan veriler trendi temsil eder. LOESS tekrar uygulanarak uzun vadeli eğilim düzleştirilir. Trend ve mevsimsel bileşen çıkarıldıktan sonra geriye kalan rastgele hata belirlenir. Zabbix'te `trendstl` fonksiyonu bir key ya da trigger expression içinde şu şekilde kullanılabilir.

`trendstl("item.key", period, "trend"|"seasonal"|"residual", interval)`

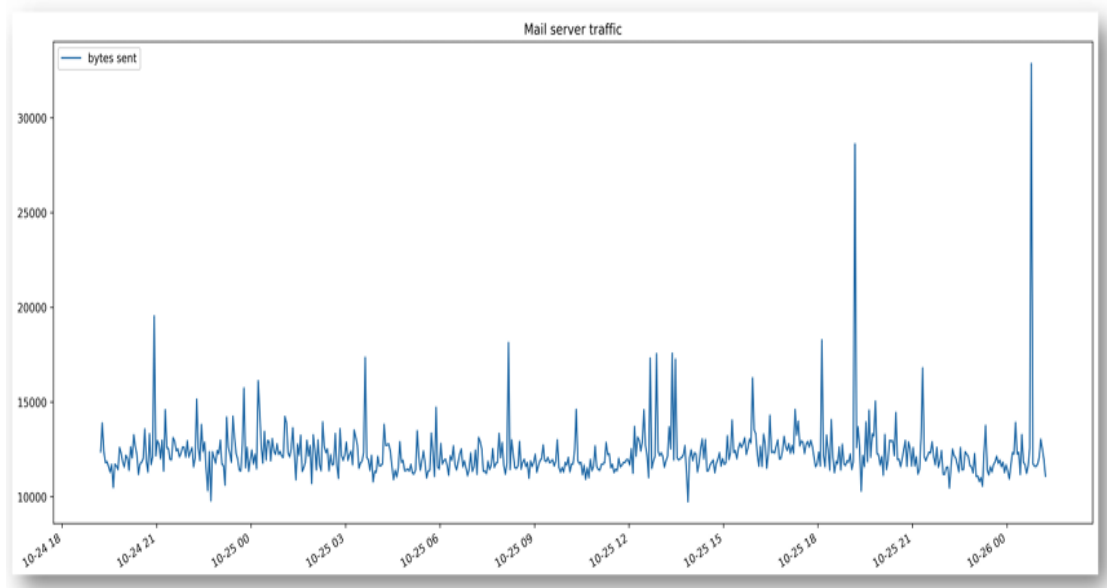
item.key: İzlenen öğenin anahtarı.

period: Mevsimsel döngünün uzunluğu.

"trend", "seasonal", "residual": Hangi bileşenin elde edileceği.

interval: Analiz edilecek zaman aralığı.

Örneğin; `trendstl("net.if.in[eth0]", 7d, "trend", 30d)` bu ifade, 30 günlük verilerdeki 7 günlük bir döngüyü kullanarak trafik eğilimini analiz eder. Avantaj olarak zaman serisi verilerindeki uzun vadeli ve dönemsel desenleri kolayca ayırır. Rastgele hataları filtreleyerek daha net bir analiz sağlar. Gürültü azaltma işlemi, STL algoritmasının ayrıştırma aşamalarında gerçekleşir. Özellikle LOESS (Locally Estimated Scatterplot Smoothing) yöntemiyle veri üzerinde lokal regresyon uygulanırken, veri setindeki rastgele sapmalar (gürültü) minimize edilir. Veriler küçük pencere aralıklarında analiz edilir ve bu pencerelerdeki noktalar bir eğriyle (yerel regresyon) modellenir. Bu yöntem, gürültü gibi kısa vadeli dalgalanmaları düzleştirerek temel trendi ve mevsimsel yapıyı vurgular. Mevsimsel desenler, belirli bir periyoda bağlı olarak yakalanır. Gürültü, bu desenlerin dışında kalan anormal değerler olarak tanımlanır ve dışlanır. Trend ve mevsimsel bileşenler çıkarıldıktan sonra kalan veri, sadece rastgele sapmaların bulunduğu artık bileşeni temsil eder. LOESS, yüksek frekanslı (ani, düzensiz) verileri düzleştirir. Bu, düşük frekanslı (trend ve sezon) sinyalleri ön plana çıkarır. Bu süreçte gürültünün doğrudan temizlenmesi değil, analiz edilen verinin bileşenlere ayrılmasıyla gürültünün etkisiz hale getirilmesi hedeflenir. Bu, verideki anlamlı desenlerin daha net görülmesini sağlar.

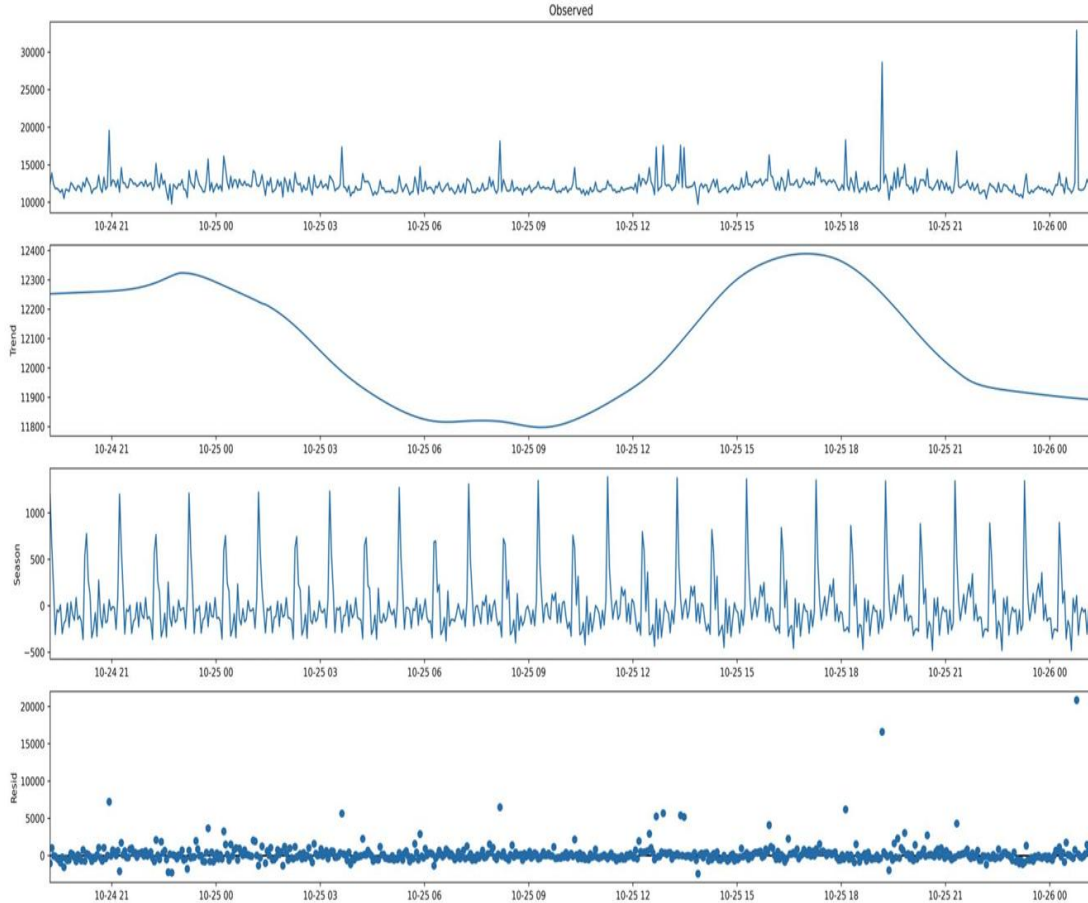


Şekil 4.1. Mail Server Trafik Örneği (Zabbix Zirvesi, 2021)

$$Y_t = T_t + S_t + R_t$$

Stl Decomposition (STL Ayırıştırma)

Orijinal Eğriyi Yumuşatma (Tt'nin Bulunması) için zaman serisi verisine yumuşatma uygulanarak temel trend (Tt) çıkarılır. Bu, uzun vadeli değişiklikleri temsil eder. Mevsimsel Bileşenlerin Ayırıştırılması (St'nin Bulunması) için orijinal seriden Tt çıkarılarak mevsimsel bileşenlerin (St) analizi yapılır. Veriler sezonlara bölünür ve her sezonun ortalaması alınır. Artıkların Hesaplanması (Rt'nin Bulunması) içinde trend (Tt) ve mevsimsel bileşenler (St) çıkarıldıktan sonra geriye sadece rastgele sapmalar (Rt) kalır. Bu artık bileşen gürültü veya anormal değerleri temsil eder. Bu şekilde zaman serisi, üç bileşene (trend, mevsimsel desen, artık) ayrılmış olur (Zabbix Zirvesi, 2021).



Şekil 4.2. Trend, Season ve Resid Grafikleri (Zabbix Zirvesi, 2021)

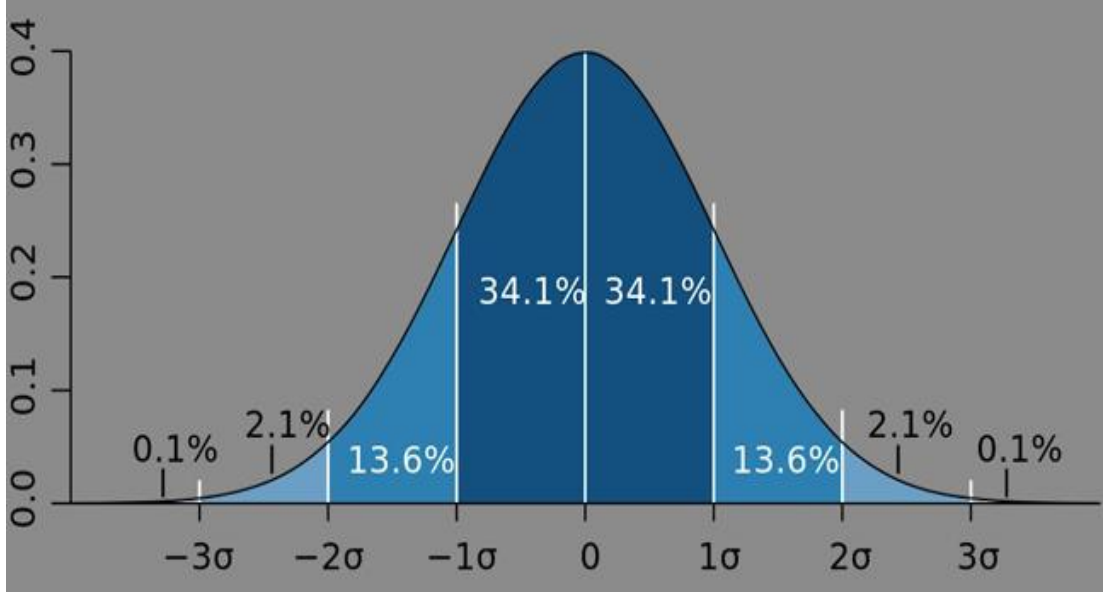
4.9.2.1. Deviations (sapmalar)

Sapmalar, verilerin ortalamadan ne kadar uzaklaştığını ölçen bir kavramdır. Zabbix'te sapmalar, verilerin değişkenliğini anlamak için kullanılır. Zabbix'te standart sapma ve medyan sapma gibi farklı sapma türleri kullanılır.

Bunlara örnek vermek gerekirse `stddevpop()` popülasyonun standart sapmasını, `stddevsamp()` örneklem standart sapmasını ve `mad()` medyan mutlak sapmasını hesaplar. Anomali Tespit Algoritması da trend değerlerini alıp, değerleri ayırıştır. Sapmaları hesaplayıp, sapmaları eşik değerine göre seçer. `Trendstl` fonksiyonu aşağıdaki şekilde kullanılabilir.

trendstl(/host/key,period:time shift,detection period,season,deviations,dev algorithm)

Bu fonksiyon, belirli bir zaman diliminde anomali tespiti yapar ve 0 ile 1 arasında bir değer döner (anomalilerin sayısının toplam değer sayısına oranını hesaplar).



Şekil 4.3. Sapma Grafik Örneği (Zabbix Zirvesi, 2021)

Şekil 4.3' te görüleceği gibi, normal dağılım (Gaussian distribution) veya çan eğrisi olarak bilinen istatistiksel dağılımı göstermektedir ve Zabbix'te makine öğrenmesi bazlı anomali tespitinde kullanılan standart sapma (standard deviation) konseptini buna göre çalışmaktadır (Zabbix Zirvesi, 2021). Şekil 4.3' teki ana noktalar için; Merkez (0): Normal sistem davranışının ortalamasını temsil eder. Standart Sapma (σ) Aralıkları:

$\pm 1\sigma$: Verilerin %68.2'si ($34.1\% + 34.1\%$)

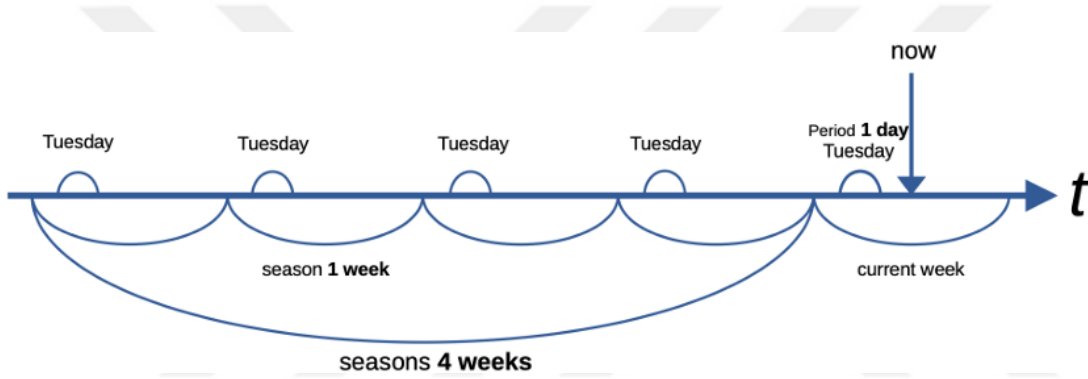
$\pm 2\sigma$: Verilerin %95.4'ü (ek $13.6\% + 13.6\%$)

$\pm 3\sigma$: Verilerin %99.8'i (ek $2.1\% + 2.1\%$)

Zabbix ML kontekstinde bu dağılım üçe ayrılır. Normal sistem davranışı için merkez etrafındaki $\pm 1\sigma$ içinde kalan değerler normal sistem davranışını gösterir ve bu aralıktaki değerler için alarm üretilmez. Potansiyel anomaliler için $\pm 2\sigma$ aralığındaki değerler "uyarı" seviyesinde anomalileri gösterebilir ve bu bölgedeki sapmalar için düşük öncelikli alertler oluşturulabilir. Kritik anomaliler $\pm 3\sigma$ dışındaki değerler (%0.1) ciddi anomalileri işaret eder ve bu bölgedeki sapmalar için yüksek öncelikli alarmlar oluşturulur. Bu istatistiksel model, Zabbix'in ML tabanlı izleme sisteminde normal sistem davranışının belirlenerek, anomali eşiklerinin otomatik ayarlanmasında bunula birlikte alert seviyelerinin belirlenmesinde ve en önemlisi proaktif izleme politikalarının oluşturulmasında kullanılır.

4.10. Baseline' a Dayalı İzleme Teknikleri

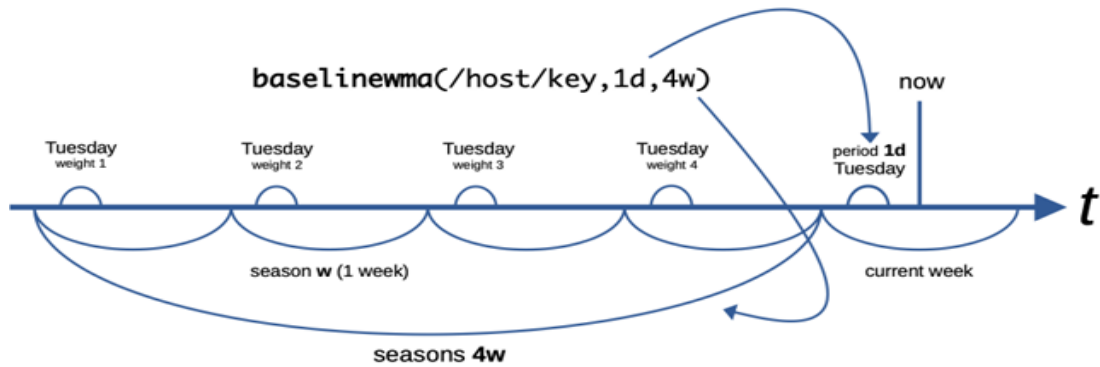
Zabbix'teki baseline (temel çizgi) fonksiyonları, geçmiş verileri detaylı bir şekilde analiz ederek, mevcut verilerin normal sınırlar içinde olup olmadığını belirlemek için kullanılır. Bu fonksiyonlar, zaman serisi verilerindeki trendleri ve mevsimsel desenleri dikkate alarak, anormallikleri tespit etmeye ve sistem performansını optimize etmeye yardımcı olur. Böylece, sistem yöneticileri ve mühendisler, olası sorunları önceden fark ederek proaktif önlemler alabilirler. Zabbix'te de `baselnewma` (Weighted Moving Average) fonksiyonu, bir zaman serisi verisinin ağırlıklı hareketli ortalamasını hesaplamak için kullanılır. Bu yöntem, geçmiş verilere dayalı olarak bir "baz çizgisi" oluşturur ve en güncel verilere daha fazla ağırlık vererek değişimlere hızlı tepki verir.



Şekil 4.4. Periyot ve Season Gösterimi (Zabbix Zirvesi, 2021)

4.10.1. Çalışma prensibi

Zaman serisi verileri (örneğin CPU kullanımı) alınır. Ağırlıklandırma olarak veriler, zaman ilerledikçe azalan bir ağırlıkla çarpılır. Son verilere daha fazla önem verilmesi için ağırlık katsayıları atanır. Ortalama hesaplama ile ağırlıklandırılmış değerlerin toplamı, ağırlıkların toplamına bölünerek ağırlıklı ortalama bulunur. Bu sonuçlar baz çizgisini temsil eder ve mevcut veriler bu çizgiyle karşılaştırılır.



Şekil 4.5. Baseline Oluşturma (Zabbix Zirvesi, 2021)

4.10.2. Algoritma detayları

BaselineWMA (Weighted Moving Average) algoritması, belirli bir metrik için zaman serisi verilerinin ağırlıklı ortalamasını alarak bir baz (baseline) değeri oluşturur. Bu algoritma, özellikle kısa ve orta vadeli trendlerin hesaplanması ve anomalilerin tespit edilmesi için uygundur. Zabbix'teki implementasyonu, geçmiş değerleri analiz eder ve bu değerlerin zaman içindeki ağırlıklarını kullanarak bir tahmin sağlar. Belirli bir zaman aralığı (örneğin, son 1 saat) için izleme verileri alınır. Bu veriler, seçilen dönemler (seasons) boyunca düzenlenir. Ağırlıkların belirlenmesi içinde Her bir veri noktası için bir ağırlık atanır. Daha yeni veriler daha yüksek ağırlık alırken, eski veriler daha düşük ağırlık alır. Ağırlıklar genellikle lineer veya geometrik bir şekilde düşürülür. Örneğin: En son veri $W=n$ (yüksek ağırlık), önceki veri $W=n-1$ şeklinde devam eder. Ağırlık Formülü aşağıdaki gibidir. Her bir veri noktası, ilgili ağırlık ile çarpılır. Ağırlıklı değerler toplanır ve toplam ağırlık değerine bölünerek hareketli ortalama hesaplanır:

$$WMA = \frac{\sum_{i=1}^n (V_i \cdot W_i)}{\sum_{i=1}^n W_i}$$

V_i : *i. zamandaki veri*

W_i : *i. zamandaki ağırlık*

n: *Değer sayısı*

Eğer "seasons" parametresi belirtilmişse, geçmiş dönemlerdeki benzer zaman dilimlerinin ortalaması alınır (örneğin, her Pazartesi saat 10:00). Bu sezonluk değerler, mevcut hareketli ortalamaya eklenerek daha doğru bir baz değeri elde edilir. Hesaplanan değer bir "baseline" olarak döndürülür ve gerçek zamanlı verilerle karşılaştırılarak anomaliler tespit edilebilir. Anormal veri dalgalanmalarını belirlemek ve trend değişimlerini daha hassas şekilde yakalamak. Bununla birlikte performans ölçümlerinde daha güncel verilerle daha doğru kararlar almayı sağlar. Bu fonksiyon, özellikle kısa dönemli tahminler ve anomalilerin tespiti için idealdir. Basit bir yapılandırmayla, geçmiş verilerle kıyaslama yaparak sistem kararlılığının sağlanması hedeflenmektedir.

baselinewma(/host/key, period<:time shift>, seasons)

/host/key: İzlenen veri ögesinin anahtar adı

period: Analiz edilecek dönemin süresi (ör. 1h, 1d)

time shift (isteğe bağlı): Verinin geçmişteki hangi zaman aralığının dikkate alınacağını belirtir

seasons: Kaç sezondan veri alınacağı. Örneğin, haftalık analiz için önceki 4 hafta kullanılabilir

Fonksiyon, belirttiğiniz dönem boyunca (ör. her Pazartesi) geçmiş verileri alıp daha sonra bu veriler, geçmiş dönemlerdeki ağırlıklarına göre bir hareketli ortalama ile değerlendirilir. Her dönem ve sezonun ortalamaları birleştirilerek baz (baseline) hesaplanır. Örneğin ;

baselinewma(/server/cpu.util,1h,10d)

Bu örnek, son 10 gün boyunca her saat için ağırlıklı bir hareketli ortalama döndürür ve baz değeri olarak kullanır. Örneğin, CPU kullanımında sürekli bir artış varsa, bu baz değere göre değerlendirilip anomali algılanabilir. Bu fonksiyon özellikle uzun dönemli analizler için uygundur. Geçmişteki ekstrem değerler baz hesaplamalarına dahil edilir. Bu durum, aşırı uçların sonuçları etkileyebileceği anlamına gelir ve doğru sonuçlar için geçmiş trend verilerinin düzenli olarak saklanması gerekir.

4.11. İleriye Yönelik Tahmine Dayalı İzleme Teknikleri

Bu çalışmada, tahmine dayalı izleme teknikleriyle, veri merkezi sistemlerinin gelecekteki davranışları ve potansiyel arızalar önceden tahmin edilecektir. Zaman serisi analizleri kullanılarak, sistemlerin performans metrikleri geleceğe yönelik olarak analiz edilecek ve bakım ihtiyaçları proaktif olarak belirlenecektir. Bu yaklaşım, sistemlerin kesinti sürelerini minimize ederek operasyonel verimliliği artıracaktır. Zabbix'in sunduğu Timeleft ve Forecast fonksiyonların kullanılarak çalışma gerçekleştirilecektir.

4.11.1. Çalışma prensibi

Timeleft fonksiyonu, bir metrik değeri ile belirli bir eşik değeri arasında kalan sürenin tahmin edilmesi için kullanılır. Örneğin, bir disk alanının dolma süresini tahmin etmek veya bir CPU yükünün belirli bir eşik seviyesine ulaşması için gereken süreyi hesaplamak amacıyla kullanılabilir. Timeleft fonksiyonu, zaman serisindeki mevcut eğilimleri analiz eder ve şu prensiplere dayanır:

Lineer Regresyon: Verinin geçmişteki davranışına göre, gelecekteki eğilimin doğrusal bir şekilde devam edeceği varsayılır.

Veri Eğilimleri: Fonksiyon, son dönemdeki verileri analiz ederek bir eğilim (trend) çıkarır. Bu eğilim doğrultusunda, eşik değeriyle kesişim noktasına ulaşmak için gereken süre hesaplanır.

Forecast fonksiyonu, zaman serisindeki mevcut verilere dayanarak gelecekteki bir zaman noktasındaki değeri tahmin etmek için kullanılır. Bu, özellikle CPU yükü, bellek kullanımı veya ağ trafiği gibi değişkenlerin gelecekteki durumunu tahmin etmek için faydalıdır. Forecast fonksiyonu, geçmiş verilere dayalı olarak bir tahmin modeli oluşturur ve şu prensiplere dayanır:

Veri Yumuşatma (Smoothing): Geçmiş verilere dayanarak ani dalgalanmaların etkisini azaltır.

Trend ve Sezonluk Bileşenler: Verideki uzun vadeli eğilimleri ve düzenli döngüleri analiz eder.

Tahmin Edilen Değer: Kullanıcının belirttiği bir zaman diliminde beklenen değeri hesaplar.

4.11.2. Algoritma detayları

Timeleft fonksiyonunda algoritma detayları için, girdi verisi olarak belirli bir izleme ögesinin (örneğin, disk kullanımı) zaman serisi verilerini alır. Zaman serisinde doğrusal bir eğilim ile trend analizi yapar. Trend doğrultusunda belirlenen eşik değeriyle mevcut veri değerinin kesişim süresini tahmin eder. Kalan süreyi dakika, saat veya gün cinsinden geri döndürür. Örnek kullanımı şu şekildedir: *timeleft(/host/disk.used,95)*, *host: Cihaz*. Bu örnek, disk kullanımının %95'e ulaşması için tahmin edilen süreyi döndürecektir. Forecast fonksiyonunda algoritma detayları ele alındığında girdi verisi olarak izleme ögesinin geçmiş zaman serisi verilerini alır. Zaman serisindeki trend ve sezonluk döngüleri analiz eder. Lineer regresyon veya diğer istatistiksel yöntemlerle tahmin modeli oluşturur. Belirtilen zaman diliminde, oluşturulan modele dayanarak beklenen değeri hesaplar ve tahmin edilen değeri döndürür. Örnek kullanımı şu şekildedir: *forecast(/host/cpu.load,10m)*, *host: Cihaz*. Bu örnek, CPU yükünün 10 dakika sonraki tahmin edilen değerini döndürecektir. Tablo 4.2' de timeleft ve forecast fonksiyonlarının karşılaştırması yapılmıştır.

Tablo 4.2. Timeleft ve Forecast Karşılaştırması:

Özellik	Timeleft	Forecast
Amaç	Eşik değerine ulaşmak için kalan süreyi bulur	Belirli bir zamandaki tahmini değeri verir
Girdi	İzleme ögesinin mevcut ve geçmiş verileri	Geçmiş veriler ve tahmin süresi
Sonuç	Süre (dakika, saat, gün.	Tahmin edilen değer
Kullanım Alanı	Disk dolma süresi, CPU yükü eşiği tahmini	Gelecekteki CPU, bellek veya ağ durumu

Timeleft fonksiyonu sayesinde kritik sistem bileşenlerinin dolma veya aşırı yüklenme gibi durumları önceden öngörülerek bakım planlaması yapılabilir. Forecast fonksiyonu ise, gelecekteki kaynak ihtiyaçlarını tahmin ederek, sistem kaynaklarının etkin bir şekilde planlanmasını sağlar. Her iki fonksiyonda, sistem performansını optimize ederek beklenmeyen kesintilerin önlenmesine yardımcı olur. Gelecekteki davranışların tahmini, daha hızlı ve akıllı kararlar alınmasını sağlar. Bu fonksiyonlar, tahmine dayalı izleme sistemlerinin temel taşlarını oluşturarak, veri merkezi operasyonlarının güvenilirliğini ve verimliliğini artıracaktır.

4.12. Otomatik İzleme ve Uyarı Sistemleri

Geliştirilen izleme sistemleri, otomatik uyarı mekanizmalarını içerecek şekilde tasarlanacaktır. Bu sayede, sistemdeki anormallikler tespit edildiğinde, insan müdahalesine gerek kalmadan otomatik olarak uyarılar gönderilecektir. Bu tür otomatik uyarılar, insan müdahalesine duyulan ihtiyacı en aza indirerek sistem yönetiminin daha verimli, güvenilir ve kesintisiz bir şekilde gerçekleştirilmesine olanak tanır. Otomatik İzleme ve Uyarı Sistemleri şu şekilde çalışır; Zabbix' e cihazlar autoregister yöntemiyle veya discovery ile otomatik olarak eklenir. Şablonlarda otomatik eklenecek şekilde tanımlanır. Bu şablonlarda sistem verileri tanımlıdır ve verileri otomatik ayarlara göre almaya başlar. Sistemden gelen metrikler ve loglar düzenli olarak toplanır. Bu veriler veritabanında tutulur. Belirli metrikler için eşik değerler ve kurallar tanımlanır (örneğin, CPU kullanımı %90'ı geçerse uyarı gönder).

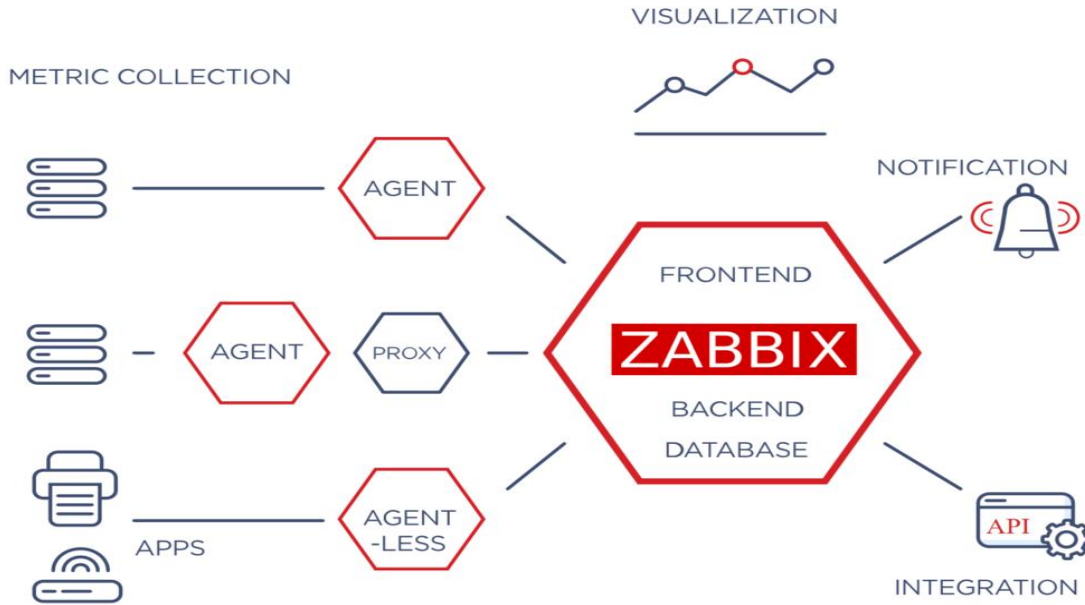
Trigger bölümünde yine şablon üzerinden gelen otomatik trigger tanımları eklidir. Gelen verilerin gerçek zamanlı analizi ile bu eşiklerin aşılması durumunda yani trigger koşulunun sağlanması ile sistem otomatik olarak anomalileri algılar. Sorun tespit edildiğinde, problem ekranına alarmlar yansır. E-posta, SMS, anlık mesajlaşma uygulamaları (ör. Telegram, Slack), veya entegrasyon yoluyla üçüncü taraf sistemlere otomatik bildirimler gönderilir. Böylelikle baştan sona otomatize bir yapı kendini genişletebilir ve otomatize şekilde çalışır hale gelir.

4.13. Deneysel Ortam ve Veri Seti

Çalışmada kullanılacak veri seti, En az bir ay boyunca ApplicationServer sunucusundan toplanan veri metriklerinden oluşmaktadır. Deneysel ortam, Open Source Zabbix izleme teknolojileri kullanılarak oluşturulmuş bir veri merkezi üzerinde kurulacaktır. Sistemin değerlendirilmesi için kullanılacak veri seti için veri toplama periyodu 1 dakikada bir ve en az bir ay boyunca metrik sayısı toplanacak ve kaydedilecektir. Sistemin doğruluğu ve etkinliği, çeşitli test senaryoları ile değerlendirilecektir. Testler, veri merkezi ortamlarında gerçekleştirilecek ve sistemin başarımlar ölçütleri (örneğin, tahmin süresi, erken uyarı etkinliği) belirlenecektir. Bu testlerin sonuçları, sistemin veri merkezi operasyonlarına katkısını ortaya koyacak ve çalışmanın hipotezinin geçerliliği üzerinde durulacaktır. Test senaryolarının detaylı tanımları ve karşılaştırmalı analizleri şeklinde değerlendirmeler yapılacaktır.

4.14. Uygulama Mimarisi

Sistem mimarisi, Zabbix izleme platformunun yeteneklerini genişletecek şekilde, makine öğrenmesi ile güçlendirilmiştir. Bu mimari, gerçek zamanlı veri toplama ve işleme yeteneklerini optimize ederek, proaktif izleme ve tahmine dayalı analizler sunar. Şekil 4.6 'da sistemin genel çalışma prensibi belirtilmiştir. Temel bileşenler arasında veri toplama katmanı, makine öğrenmesi algoritmalarını barındıran analiz motoru, kullanıcı dostu bir gösterge paneli ve sistem yöneticilerine erken uyarı gönderen bildirim modülleri yer almaktadır. Ayrıca, ölçeklenebilir bir altyapı ve güvenlik mekanizmaları da tasarımın ayrılmaz bir parçasıdır.



Şekil 4.6. Zabbix Sistem Mimarisi (Evonsys, n.d.)

4.14.1. Zabbix server uygulaması

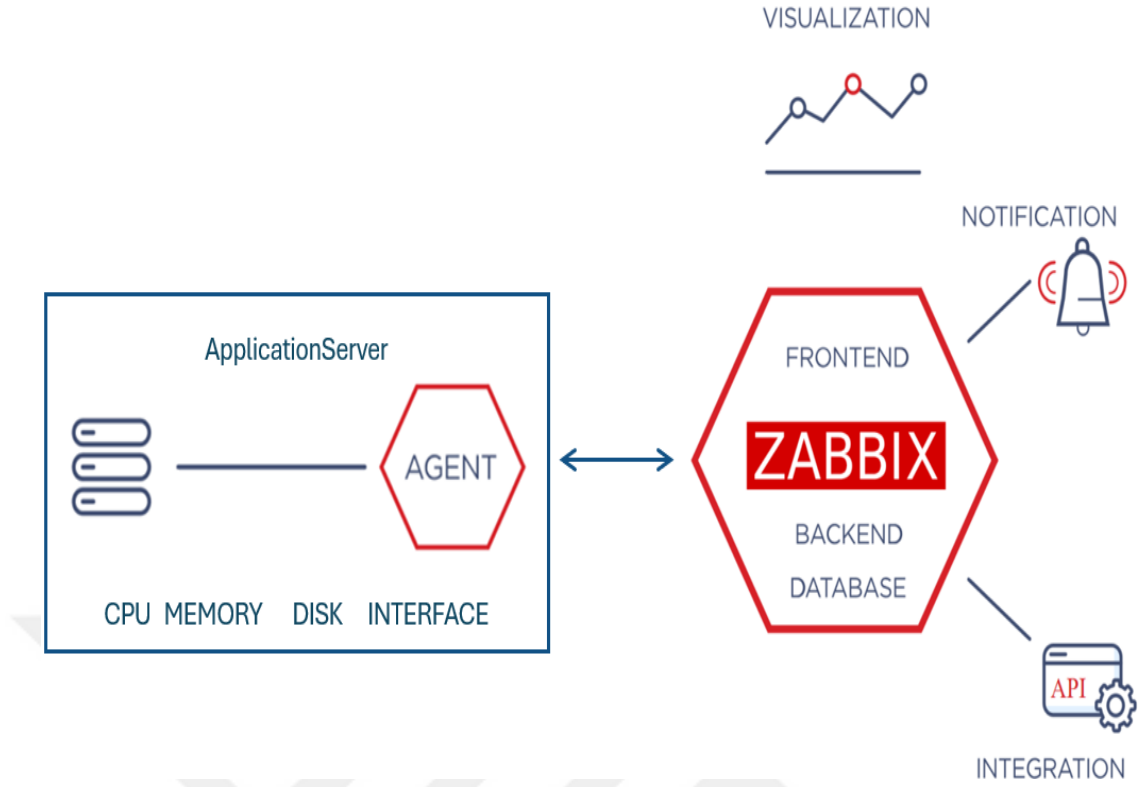
Zabbix izleme altyapısının merkezi bileşenidir. Tüm izleme verilerinin toplandığı, işlendiği ve saklandığı bir sunucu yazılımıdır. Zabbix Server, ajanlar, proxy'ler ve API'ler aracılığıyla izlenen sistemlerden veri alır ve bu verilerle analizler yaparak kullanıcıya anlamlı bilgiler sunar. Aynı zamanda, tetikleyiciler aracılığıyla alarmlar oluşturur ve kullanıcıları sistemdeki kritik durumlar hakkında bilgilendirir.

4.14.2. Zabbix agent uygulaması

Zabbix sunucusuna veya proxy'ye veri toplamak için kullanılan hafif bir yazılımdır. İzlenecek olan sistemlere (örneğin sunucular, sanal makineler veya masaüstü bilgisayarlar) yüklenir ve bu sistemlerden performans, durum bilgisi ve çeşitli metrikleri toplar. Zabbix Agent, Zabbix izleme altyapısının önemli bir bileşenidir ve gerçek zamanlı izleme için temel verileri sağlar.

4.14.3. Veri toplama servisi

Zabbix ile bir çok yöntem kullanılarak veri toplama yapılabilir. Bu çalışmada Zabbix Agent'lar aracılığıyla Linux sunuculardan veri toplama işlemi gerçekleştirilecektir. Şekil 4.7' de genel olarak izleme şeması belirtilmiştir. Linux sistemlerden toplanacak temel metrikler; CPU kullanımı, Memory kullanımı, Disk kullanımı ve Ağ I/O trafiğidir. Özelleştirilmiş ML template'leri ile de metrik toplama ve metrikleri işleme gerçekleştirilecektir.



Şekil 4.7. Zabbix ile ApplicationServer Sunucu İzlemesi (Evonsys, n.d.)

4.14.4. Ön işleme ve değerlendirme

Ön işleme ve değerlendirme yapabilmek adına toplanan verilerin doğru türde saklanması sağlanır. Örneğin text, log, numeric gibi türler ilk item tanımı yapılırken belirlenir. Biz numeric değerleri kullanacağız. Zabbix veritabanında depolanan veriler ML algoritmaları ile kullanılabilmesi adına Trigger tanımlarından ML algortima fonksiyonları kullanılacaktır.

4.14.5. Uyarı ve bildirim sistemi

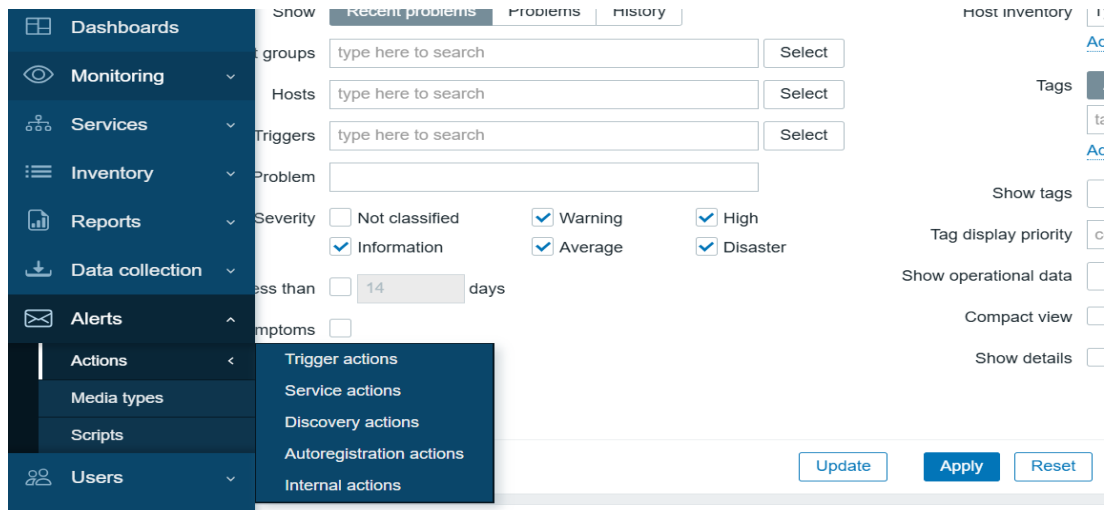
Zabbix'in native alert sistemi ile ML tahminlerinde kullanılacak trigger yapıları eşleştirilir. Burada oluşacak herhangi bir alarm problem ekranına yansır ve burada actions olarak tanımlanmış veri setlerine göre bu uyarı ilgili noktalara ulaştırılır. Böylelikle anında bildirimler ile sorunlara müdahale en hızlı noktaya getirilmiş olacaktır. Bu sayede, veri merkezlerinin iş sürekliliği kesintiye uğramadan sağlanırken, yöneticiler ve teknik ekipler daha stratejik bir şekilde hareket edebilmektedir.



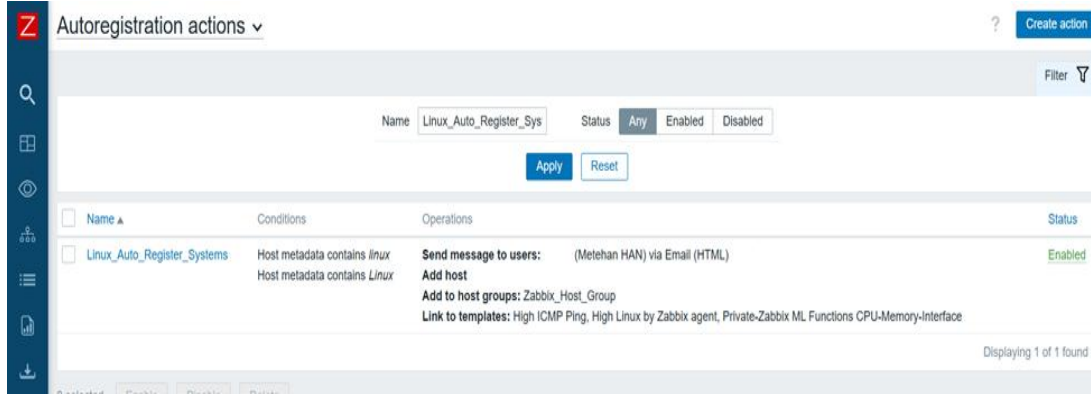
5. UYGULAMA VE TEST

Çalışmanın bu bölümünde Zabbix Uygulaması üzerinde izleme sistemlerinin nasıl kullanıldığı, sistem tasarımının nasıl oluşturulduğu; reaktif ve proaktif izlemelerin değerlendirilmesi sağlanacaktır. Özellikle makine öğrenmesi algoritmaları kullanımını ele alarak süreci uçtan uca nasıl tasarlayıp pratik hale getirdiğimizi ve bu yapıda testlerimizi veriler üzerinde nasıl gerçekleştirildiği gösterilecektir.

Zabbix uygulaması Linux sunucu üzerinde Zabbix dökümanlarına göre kurulmuştur. Açık kaynak PostgreSQL veritabanına sahiptir. Zabbix' e yönetici kullanıcımız ile giriş yaparak Sol bölümde bulunan Data Collection bölümüne gidilir. Burada öncelikle Auto Register bölümünden linux sunucuların bulunabileceği tanımlar sağlanır. Auto Register yapısı şu şekilde çalışır: Zabbix Agent'lar izleyeceğimiz sunucular üzerine kurulur. Burada Zabbix Agent conf dosyasında HostMetadataItem=system.uname parametresi eklenir. Agent restart edilir. Zabbix uygulama arayüzünde Actions menüsünden Autoregistration bölümüne aşağıdaki görselde olduğu gibi tanımlar yapılır. Böylelikle Zabbix her iki dakika da bir kontrol ederek sisteme dahil olan yeni bir sunucu var ise onu bulur ve otomatik olarak Zabbix' e dahil eder. Autoregister kısmında eklenecek sunucuya neler dahil edileceği de belirtilebilir. Linux sunucu izleme şablonu ApplicationServera otomatik olarak dahil edilir. Bununla birlikte ML için oluşturulan şablonlar da dahil edilir.



Şekil 5.1. Alarm Bölümü Autoregistration Ayarlanması



Şekil 5.2. Linux Auto Register Tanımı

Action

Action Operations 4

* Name

Type of calculation A or B

Conditions	Label	Name	Action
A		Host metadata contains <i>linux</i>	Remove
B		Host metadata contains <i>Linux</i>	Remove

[Add](#)

Enabled ☐

* At least one operation must exist.

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Şekil 5.3. Auto Register Şartlarının Belirlenmesi

Action

Action Operations 4

Operations

Details	Action
Send message to users: (Metehan HAN) via Email (HTML)	Edit Remove
Add host	Edit Remove
Add to host groups: Zabbix_Host_Group	Edit Remove
Link to templates: High ICMP Ping, High Linux by Zabbix agent, Private-Zabbix ML Functions CPU-Memory-Interface	Edit Remove

[Add](#)

* At least one operation must exist.

[Update](#) [Clone](#) [Delete](#) [Cancel](#)

Şekil 5.4. Şartlar Oluşturduğunda Alınacak Aksiyonlar

Bundan sonraki süreçte host denilen yani sunucular üzerinden özel olarak eklemek istenilen tanımları da Data Collection menüsünden Hosts kısmından istediğiniz sunucuyu aratarak o sunucu üzerinde Item bölümüne giriş yapılır.

Burada Item bu sunucu üzerinde veri toplamak için tanımlanan ve toplanan veriye ulaşmamızı sağlayan bir parametredir. Zabbix Agent vasıtası ile bir çok değere erişim sağlanabilir.

Tablo 5.1. Zabbix İtem Veri İzleme Tanımları

Item Name	Item Key	Interval
CPU utilization	system.cpu.util	1m
Memory utilization	vm.memory.utilization	1m
CPU Util Forecast in 3h analysis to 15m value	cpu.calc.item.util.forecast	1m
Memory Util Forecast in 3h analysis to 15m value	memory.calc.item.util.forecast	1m
Interface Receive Util Low Calc Item	interface.receive.calc.item.low.util	5m
Interface Sent Util High Calc Item	interface.sent.calc.item.high.util	5m
/: Space utilization	vfs.fs.size[/,pused]	1m

Çalışma için CPU, Memory, Ağ I/O ve Disk verileri önemli. Bu verilerin izlenmesi ApplicationServer sunucusu üzerindedir. Bu veriler için ApplicationServer sunucusu üzerinde Tablo 5.1’ de görülen örnek tanımlar yapıldı. Bu tanımlar üzerinden istenilen veriler alınarak Zabbix veritabanında analiz edilmek için saklanacaktır.

☐	Severity	Value	Name ▲	Operational data
☐	Average	OK	Mounted filesystem discovery: --ANOMALY-- / : Disk kapasitesi 7 gün içerisinde (used > {\$VFS.FS.PUSED.MAX.CRIT:""}%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})
☐	Average	OK	Mounted filesystem discovery: --ANOMALY-- /home: Disk kapasitesi 7 gün içerisinde (used > {\$VFS.FS.PUSED.MAX.CRIT:"/home"}%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})
☐	Average	OK	Mounted filesystem discovery: --ANOMALY-- /opt/repositories: Disk kapasitesi 7 gün içerisinde (used > {\$VFS.FS.PUSED.MAX.CRIT:"/opt/repositories"}%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})
☐	Average	OK	Mounted filesystem discovery: --ANOMALY-- /tmp: Disk kapasitesi 7 gün içerisinde (used > {\$VFS.FS.PUSED.MAX.CRIT:"/tmp"}%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})
☐	High	OK	Mounted filesystem discovery: / : Disk space is critically low (used > {\$VFS.FS.PUSED.MAX.CRIT:""}% and available space < 30GB)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})
☐	Warning	OK	Mounted filesystem discovery: / : Disk space is low (used > {\$VFS.FS.PUSED.MAX.WARN:""}%) Depends on: ApplicationServer: / : Disk space is critically low (used > {\$VFS.FS.PUSED.MAX.CRIT:""}% and available space < 30GB)	Space used: {ITEM.LASTVALUE3} of {ITEM.LASTVALUE2} ({ITEM.LASTVALUE1})
☐	High	OK	Mounted filesystem discovery: / : Running out of free inodes (free < {\$VFS.FS.INODE.PFREE.MIN.CRIT:""}%)	Free inodes: {ITEM.LASTVALUE1}
☐	Not classified	OK	Private-Zabbix ML Functions CPU-Memory-Interface: High CPU utilization (over %80 for 20m)	
☐	Not classified	OK	Private-Zabbix ML Functions CPU-Memory-Interface: High CPU utilization (over %90 for 20m)	
☐	Average	OK	Linux CPU by Zabbix agent: High CPU utilization (over {\$CPU.UTIL.AVG}% for 30m) Depends on: ApplicationServer: High CPU utilization (over {\$CPU.UTIL.CRIT}% for 10m)	Current utilization: {ITEM.LASTVALUE1}
☐	High	OK	Linux CPU by Zabbix agent: High CPU utilization (over {\$CPU.UTIL.CRIT}% for 10m)	Current utilization: {ITEM.LASTVALUE1}
☐	Average	OK	Linux memory by Zabbix agent: High memory utilization (>{\$MEMORY.UTIL.AVG}% for 30m) Depends on: ApplicationServer: High memory utilization (>{\$MEMORY.UTIL.MAX}% for 10m) ApplicationServer: Lack of available memory (<{\$MEMORY.AVAILABLE.MIN} of {ITEM.VALUE2})	
☐	High	OK	Linux memory by Zabbix agent: High memory utilization (>{\$MEMORY.UTIL.MAX}% for 10m) Depends on: ApplicationServer: Lack of available memory (<{\$MEMORY.AVAILABLE.MIN} of {ITEM.VALUE2})	
☐	Not classified	OK	Private-Zabbix ML Functions CPU-Memory-Interface: -ML- Interface Receive Kullanımında Anomali : Son 15 dakikalık Interface Receive ortalaması, Son 4 haftalık Interface Receive baseline verilerine göre %20 veya daha fazladır.	Current utilization: {ITEM.LASTVALUE1}
☐	Not classified	OK	Private-Zabbix ML Functions CPU-Memory-Interface: -ML- Interface Sent Kullanımında Anomali : Son 15 dakikalık Interface Sent ortalaması, Son 4 haftalık Interface Sent baseline verilerine göre %20 veya daha azdır.	Current utilization: {ITEM.LASTVALUE1}
☐	Not classified	OK	CPU Util Forecast in {\$FORECAST_CPU_UTIL_3H} analysis to {\$FORECAST_CPU_UTIL_15M} value	min({ApplicationServer_UTIL_70})
☐	Not classified	OK	Memory Util Forecast in {\$FORECAST_MEMORY_UTIL_3H} analysis to {\$FORECAST_MEMORY_UTIL_15M} value	min({ApplicationServer_UTIL_50})

Şekil 5.5. Sunucu Üzerinde Tanımlı Örnek Alarmlar

Bu tanımların bir kısmını şablon üzerinde otomatik olarak yaparken geriye kalan kısmını da istenilen verileri değerlendirebilmek adına sunucu item bölümünden özel olarak yapıldı. Her iki yöntemde istenilen duruma göre kullanılabilir. Bu tanımlardan sonra yine şablonda tanımlanan veya özel olarak tanımlanan item'lara bağlı Tablo 5.2' de örnek olarak bulunan trigger'lar oluşturuldu.

Trigger Zabbix'te kontrol edilen ve eşik değerler, koşullar sağlanırsa alarm oluşturan yapılardır. Örnek bir şablon tanımı Şekil 5.6' daki gibidir. Şekil 5.7 ve 5.8' de ise örnek tanımlar gösterilmiştir.

Tablo 5.2. Zabbix Alarm Tanımları

Trigger Name	Expression
-ML- CPU Kullanımında Anomali : Son 15 dakikalık CPU ortalaması, Son 4 haftalık CPU baseline verilerine göre % X daha fazladır. Son 15 Dk : Y Baseline Verisi : Z	last(/ApplicationServer/system.cpu.util,#1) > 20 and (avg(/ApplicationServer/system.cpu.util,15m) > 1,3 * baselinewma(/ApplicationServer/system.cpu.util,1h:now/h,"w",4))
-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre % X daha azdır. Son 15 Dk : Y Baseline Verisi : Z	last(/ApplicationServer/vm.memory.utilization,#1) > 5 and (avg(/ApplicationServer/vm.memory.utilization,15m) / baselinewma(/ApplicationServer /vm.memory.utilization,1h:now/h,"w",4) < { \$ML_MEMORY_LOW_UTIL })
-ML- Interface Sent Kullanımında Anomali : Son 15 dakikalık Interface Sent ortalaması, Son 4 haftalık Interface Sent baseline verilerine göre % X daha fazladır. Son 15 Dk : Y Baseline Verisi : Z	last(/ApplicationServer/net.if.out["ens192"],#1) > 1000000 and (avg(/ApplicationServer/net.if.out["ens192"],15m) > 1,3 * baselinewma(/ApplicationServer/net.if.out["ens192"],1h:now/h,"w",4))
--ANOMALY-- /: Disk kapasitesi 7 gün içerisinde (used >90%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.)	timeleft(/ApplicationServer/vfs.fs.size[/,pused],24h,100)<7d and timeleft(/ApplicationServer/vfs.fs.size[/,pused],24h,100)<>-1
High memory utilization %90 for 20m	min(/ApplicationServer/vm.memory.utilization,15m)>90
High CPU utilization (over %90 for 20m)	min(/ApplicationServer/system.cpu.util,15m)>90

Templates

All templates / Private-Zabbix ML Functions CPU-... Items 22 Triggers 22 Graphs Dashboards Discovery rules Web scenarios

Template Tags Macros 8 Value mapping

* Template name Private-Zabbix ML Functions CPU-Memory-Interface

Visible name Private-Zabbix ML Functions CPU-Memory-Interface

Templates type here to search Select

* Template groups Templates/Applications x type here to search Select

Description

Update Clone Full clone Delete Delete and clear Cancel

Şekil 5.6. Zabbix Şablon Kullanımı

	Name ▼	Triggers	Key	Interval	History	Trends	Type	Status	Tags
☐	Memory Util Low Calc Item	Triggers 1	memory.calc.item.low.util	5m	90d	180d	Calculated	Enabled	Name: ML_Memory_L...
☐	Memory Util High Calc Item	Triggers 1	memory.calc.item.high.util	5m	90d	180d	Calculated	Enabled	Name: ML_Memory_H...
☐	Memory Calc Item Utilization 90	Triggers 1	memory.calc.item.util90	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Memory Calc Item Utilization 80	Triggers 1	memory.calc.item.util80	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Memory Calc Item Utilization 70	Triggers 1	memory.calc.item.util70	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Memory Calc Item Utilization 60	Triggers 1	memory.calc.item.util60	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Memory Calc Item Utilization 50	Triggers 1	memory.calc.item.util50	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Memory Calc Item Utilization 30	Triggers 1	memory.calc.item.util30	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Memory Calc Item Utilization 20	Triggers 1	memory.calc.item.util20	5m	90d	180d	Calculated	Enabled	Name: Memory_Utiliz...
☐	Interface Sent Util Low Calc Item	Triggers 1	interface.sent.calc.item.low.util	5m	90d	180d	Calculated	Enabled	Name: ML_Interface_...
☐	Interface Sent Util High Calc Item	Triggers 1	interface.sent.calc.item.high.util	5m	90d	180d	Calculated	Enabled	Name: ML_Interface_...
☐	Interface Receive Util Low Calc Item	Triggers 1	interface.receive.calc.item.low.util	5m	90d	180d	Calculated	Enabled	Name: ML_Interface_...
☐	Interface Receive Util High Calc Item	Triggers 1	interface.receive.calc.item.high.util	5m	90d	180d	Calculated	Enabled	Name: ML_Interface_...
☐	CPU Util Low Calc Item	Triggers 1	cpu.calc.item.low.util	5m	90d	180d	Calculated	Enabled	Name: ML_CPU_Low...
☐	CPU Util High Calc Item	Triggers 1	cpu.calc.item.high.util	5m	90d	180d	Calculated	Enabled	Name: ML_CPU_High...
☐	CPU Calc Item Utilization 90	Triggers 1	cpu.calc.item.util90	5m	90d	180d	Calculated	Enabled	Name: CPU_Utilizatio...

Şekil 5.7. Şablon İçerisinde Bulunan Tanımlar

	Severity	Name ▲	Operational data	Expression	Status	Tags
☐	Not classified	-ML- CPU Kullanımında Anomali : Son 15 dakikalık CPU ortalaması, Son 4 haftalık CPU baseline verilerine göre %20 veya daha azdır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/cpu.calc.item.low.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_CPU_Low...
☐	Not classified	-ML- CPU Kullanımında Anomali : Son 15 dakikalık CPU ortalaması, Son 4 haftalık CPU baseline verilerine göre %20 veya daha fazladır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/cpu.calc.item.high.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_CPU_High...
☐	Not classified	-ML- Interface Receive Kullanımında Anomali : Son 15 dakikalık Interface Receive ortalaması, Son 4 haftalık Interface Receive baseline verilerine göre %20 veya daha azdır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/interface.receive.calc.item.low.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_Interface_...
☐	Not classified	-ML- Interface Receive Kullanımında Anomali : Son 15 dakikalık Interface Receive ortalaması, Son 4 haftalık Interface Receive baseline verilerine göre %20 veya daha fazladır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/interface.receive.calc.item.high.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_Interface_...
☐	Not classified	-ML- Interface Sent Kullanımında Anomali : Son 15 dakikalık Interface Sent ortalaması, Son 4 haftalık Interface Sent baseline verilerine göre %20 veya daha azdır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/interface.sent.calc.item.low.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_Interface_...
☐	Not classified	-ML- Interface Sent Kullanımında Anomali : Son 15 dakikalık Interface Sent ortalaması, Son 4 haftalık Interface Sent baseline verilerine göre %20 veya daha fazladır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/interface.sent.calc.item.high.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_Interface_...
☐	Not classified	-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre %20 veya daha azdır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/memory.calc.item.low.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_Memory_L...
☐	Not classified	-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre %20 veya daha fazladır.	Current utilization: (ITEM.LASTVALUE1)	last(/Private-Zabbix ML Functions CPU-Memory-Interface/memory.calc.item.high.util,#1)=1	Enabled	App: Zabbix_Test Name: ML_Memory_H...
☐	Not classified	High CPU utilization (over %20 for 20m)		last(/Private-Zabbix ML Functions CPU-Memory-Interface/cpu.calc.item.util20,#1)=1	Enabled	App: Zabbix_Test Name: Cpu_Utilization...

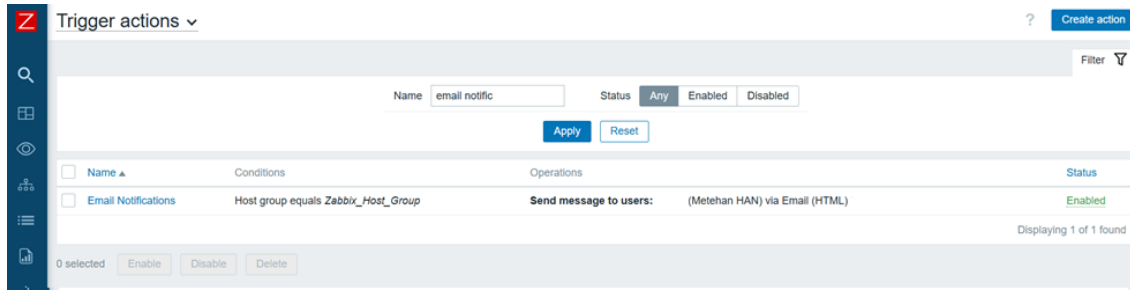
Şekil 5.8. Alarm Tanımları

Bu trigger alarmları Zabbix uygulamasında Monitoring menüsü üzerinde Problems ekranından takip edilebilir. Burada açık alarmlar ve geçmişte gelen alarmlar incelenebilmektedir. Özellikle bazı alarmlarda, verileri alarm içerisine de özel yöntemler ile yazdırıldı. Böylelikle görme ve analizler daha rahat olacaktır. Örnek alarmlar Şekil 5.9’ daki gibidir.

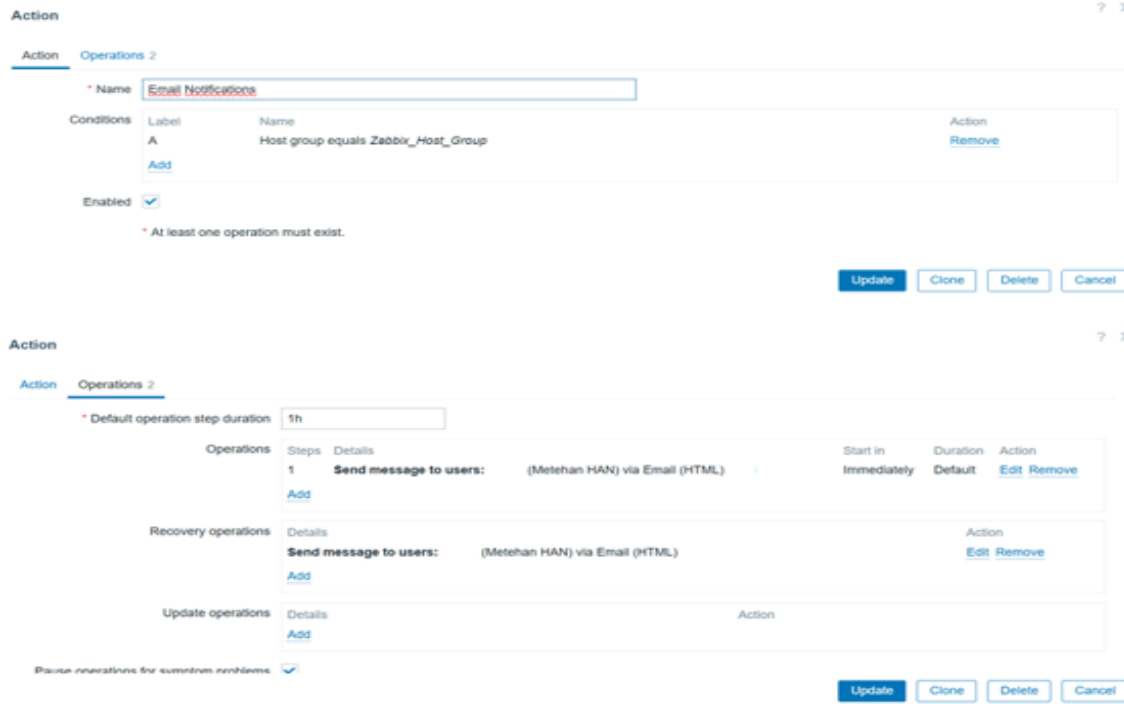
Time	Severity	Recovery time	Status	Info	Host	Problem	Operational data	Duration	Update	Actions	Tags
2024-12-06 03:16:49	Not classified	2024-12-06 04:01:49	RESOLVED		Applicatio nServer	-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre %34.64 daha azdır. Son 15 Dk : 11.27 Baseline Verisi : 17.25					
2024-12-06 02:32:11	Not classified	2024-12-06 02:42:11	RESOLVED		Applicatio nServer	-ML- Interface Sent Kullanımında Anomali : Son 15 dakikalık Interface Sent ortalaması, Son 4 haftalık Interface Sent baseline verilerine göre %56 daha fazladır. Son 15 Dk : 2482870 Baseline Verisi : 1587737					
2024-12-06 02:32:04	Not classified	2024-12-06 02:37:04	RESOLVED		Applicatio nServer	-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre %33.44 daha fazladır. Son 15 Dk : 26.47 Baseline Verisi : 19.83					
2024-12-03 02:07:07	Not classified	2024-12-03 02:12:07	RESOLVED		Applicatio nServer	-ML- Interface Receive Kullanımında Anomali : Son 15 dakikalık Interface Receive ortalaması, Son 4 haftalık Interface Receive baseline verilerine göre %51 daha azdır. Son 15 Dk : 1671766 Baseline Verisi : 3369675					
2024-12-03 02:00:53	Not classified	2024-12-03 06:34:53	RESOLVED		Applicatio nServer	Memory Util Forecast in 3h analysis to 15m value					
2024-12-03 02:00:53	Not classified	2024-12-03 06:34:53	RESOLVED		Applicatio nServer	Memory Util Forecast in 3h analysis to 15m value					
2024-12-03 01:42:00	Not classified	2024-12-03 01:47:00	RESOLVED		Applicatio nServer	High CPU utilization (over %60 for 20m)					
2024-12-03 01:05:54	Not classified	2024-12-03 02:52:54	RESOLVED		Applicatio nServer	CPU Util Forecast in 3h analysis to 15m value					
2024-11-26 16:47:01	Average	2024-11-26 23:57:01	RESOLVED		Applicatio nServer	--ANOMALY-- /var: Disk kapasitesi 7 gün içerisinde (used > 90%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.)					
2024-11-26 16:37:13	Not classified	2024-11-26 16:47:13	RESOLVED		Applicatio nServer	-ML- Interface Receive Kullanımında Anomali : Son 15 dakikalık Interface Receive ortalaması, Son 4 haftalık Interface Receive baseline verilerine göre %123 daha fazladır. Son 15 Dk : 476313 Baseline Verisi : 212741					

Şekil 5.9 Örnek Alarmlar

Data Collection menüsü üzerinde Actions bölümünde Trigger Actions menüsünden gelecek alarm yapılarına, gruplarına göre istenilen aksiyonu alması sağlanabilir. İlgili birimlere mail veya sms gönderimi sağlanabilir. Örnek bir tanım Şekil 5.10 ve 5.11’ deki gibidir.

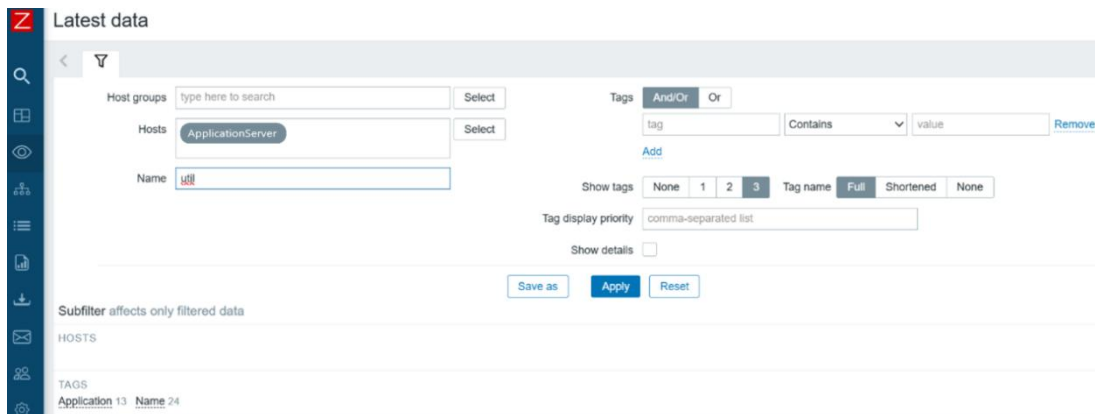


Şekil 5.10. Alarm Gönderim Tanımları



Şekil 5.11. Alarm Konfigürasyonları

İzlenen verileri de görebilmek için Monitoring menüsü altında bulunan Latest Data bölümüne gidilerek Zabbix arayüzünden kontrol edilebilir. Örnek gösterim Şekil 5.12'deki gibidir.



Şekil 5.12. İzlenen Verilerin Gösterimi

Host	Name	Last check	Last value	Change	Tags	
ApplicationServer	/ Free inodes in %	2m 1s	98.4259 %		Application Filesystem	Graph
ApplicationServer	/ Space utilization	58s	60.3191 %		Application Filesystem	Graph
ApplicationServer	/ Total space	28m 51s	9.99 GB		Application Filesystem	Graph
ApplicationServer	/ Used space	44s	6.03 GB		Application Filesystem	Graph
ApplicationServer	/home: Free inodes in %	2m 1s	99.9981 %		Application Filesystem	Graph
ApplicationServer	/home: Space utilization	57s	1.3598 %		Application Filesystem	Graph
ApplicationServer	/home: Total space	28m 50s	4.99 GB		Application Filesystem	Graph
ApplicationServer	/home: Used space	43s	68.48 MB		Application Filesystem	Graph
ApplicationServer	CPU Util Forecast in (\$FORECAST_CPU_UTIL_3H) analysis to (\$F...	47s	0.6402	-0.0006064	Name: ML_CPU_Util...	Graph
ApplicationServer	CPU Util High Calc Item	2m 44s	0		Name: ML_CPU_Hig...	Graph
ApplicationServer	CPU utilization	9s	0.664 %	+0.02936 %	Application: CPU	Graph
ApplicationServer	CPU Util Low Calc Item	3m 1s	0		Name: ML_CPU_Lo...	Graph
ApplicationServer	Interface ens192: Outbound packets discarded	2m 5s	0		Application: Interface	Graph
ApplicationServer	Interface ens192: Outbound packets with errors	2m 4s	0		Application: Interface	Graph
ApplicationServer	Interface ens224: Bits received	25s	352 bps	-16 bps	Application: Interface	Graph
ApplicationServer	Interface ens224: Bits sent	24s	352 bps	+64 bps	Application: Interface	Graph
ApplicationServer	Interface ens224: Inbound packets discarded	26s	0		Application: Interface	Graph
ApplicationServer	Interface ens224: Inbound packets with errors	26s	0		Application: Interface	Graph
ApplicationServer	Memory Calc Item Utilization 90	3m 24s	0		Name: Memory_Utiliz...	Graph
ApplicationServer	Memory Util Forecast in (\$FORECAST_MEMORY_UTIL_3H) analys...	19s	6.1822	+0.0001929	Name: ML_Memory_...	Graph
ApplicationServer	Memory Util High Calc Item	3m 16s	0		Name: ML_Memory_...	Graph
ApplicationServer	Memory utilization	1s	14.6316 %	+0.004059 %	Application: Memory	Graph
ApplicationServer	Memory Util Low Calc Item	3m 31s	0		Name: ML_Memory_...	Graph

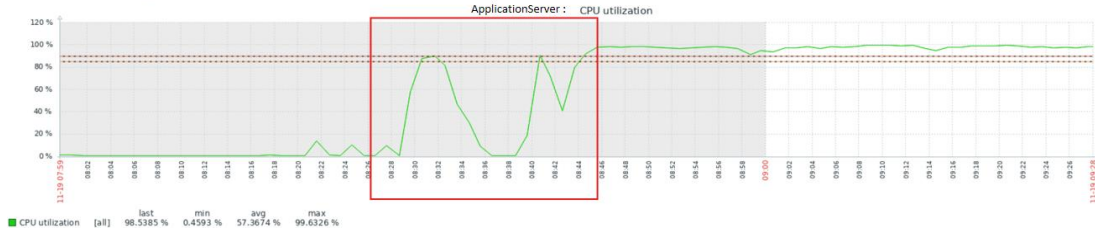
Şekil 5.12. (Devamı) İzlenen Verilerin Gösterimi

Şekil 5.12’ de yanında bulunan grafik seçeneği ile verileri geçmişe yönelik istenilen aralıklarda kontrol edilebilir. Bu çalışma için monitor edilen ApplicationServer sunucusu öncesinde bahsedilen adımlarla tanımları yapılarak izlemeye alınmıştır.

İlk senaryoda Şekil 5.14’ teki grafikte görüleceği üzere cpu kullanımları olağan devam ederken artış anlarını yakalayarak geçmişe göre değerlendirme sonrası anomali tespit edilerek Şekil 5.13’ te bulunan alarm yansımasıdır. Geleneksel izleme yönteminde alarm ise 09.04’ te yansımasıdır. Makine öğrenmesi ile 08.42 saatinde alarm yansımasıdır. Normal tanımlara göre 22 dakika önceden anomali başladığında yakalanmıştır. Burada son 4 haftanın baseline verileri baz alındığında son 15 dakikalık cpu ortalamasında anormal bir artış olduğu gözlemlenmiştir.

2024-11-19 09:04:43	High	2024-11-19 09:34:42	RESOLVED	Application Server	High CPU utilization (over 90% for 20m) ?
2024-11-19 08:42:05	Not classified	2024-11-19 09:42:05	RESOLVED	Application Server	-ML- CPU Kullanımında Anomali : Son 15 dakikalık CPU ortalaması, Son 4 haftalık CPU baseline verilerine göre %2034.87 daha fazladır. Son 15 Dk : 39.72 Baseline Verisi : 1.86 ?

Şekil 5.13. CPU Yüksek Kullanım Alarmları

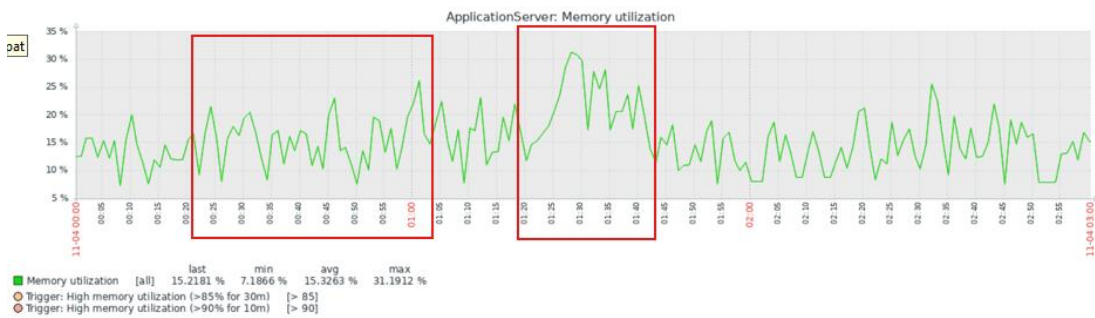


Şekil 5.14. CPU Yüksek Kullanım Grafiği

Bir başka senaryoda Şekil 5.15’ te görülen 00.32 ve 01.32 saatlerinde son 4 haftanın baseline verileri baz alındığında son 15 dakikalık memory ortalamasında anormal bir artış olduğu, ilk alarmda yaklaşık 2 kat bir artış olmuşken İkinci alarmda %40’a yakın artış gözlemlenmiştir. Geleneksel izleme yönteminde belirlenen eşik değeri aşılmadığı için alarm yansımamıştır. Şekil 5.16’ ya bakıldığında sorun tespiti yapılması zorlaşacak ve geriye yönelik grafik analizleri çalışanlar tarafından yapılmak zorunda idi. Bu da yapıda maliyet ve fazla çalışma anlamına gelmektedir.

2024-12-04 01:32:04	Not classified	2024-12-04 02:07:04	RESOLVED	Applicat ionServ er	-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre %39.84 daha fazladır. Son 15 Dk : 23.37 Baseline Verisi : 16.71
2024-12-04 00:32:04	Not classified	2024-12-04 01:02:04	RESOLVED	Applicat ionServ er	-ML- Memory Kullanımında Anomali : Son 15 dakikalık Memory ortalaması, Son 4 haftalık Memory baseline verilerine göre %98.57 daha fazladır. Son 15 Dk : 19.15 Baseline Verisi : 9.65

Şekil 5.15. Memory Yüksek Kullanım Alarmları



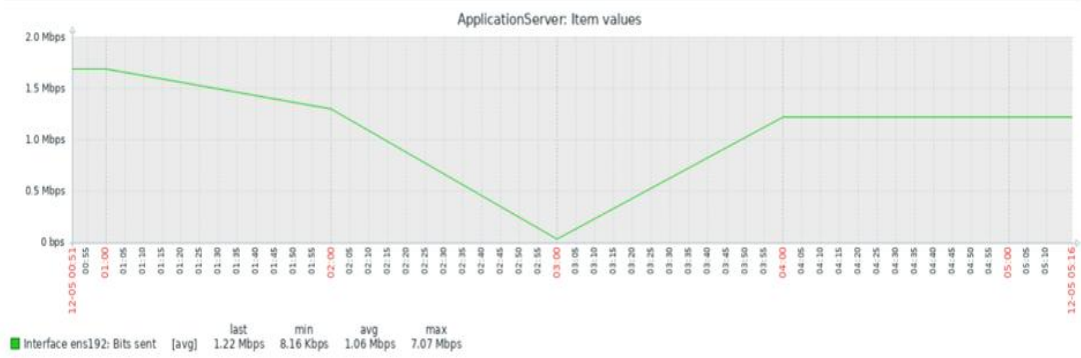
Şekil 5.16. Memory Yüksek Kullanım Grafiği

Bir başka senaryoda yine geleneksel izlemede pratik olarak çok kullanılmayan ama makine öğrenmesi sayesinde düşüşlerde takip edilebilmektedir. Şekil 5.18 ve 5.19’ daki grafiklerde görüleceği üzere sunucunun network sent ve receive değerlerindeki kullanımlar olağan devam ederken düşüş anlarını yakalayıp geçmişe göre değerlendirme sonrası anomali tespit edilerek Şekil 5.17’ de bulunan alarm yansımıştır. Böylelikle sorun tespiti yapılması ve geriye yönelik grafik analizleri

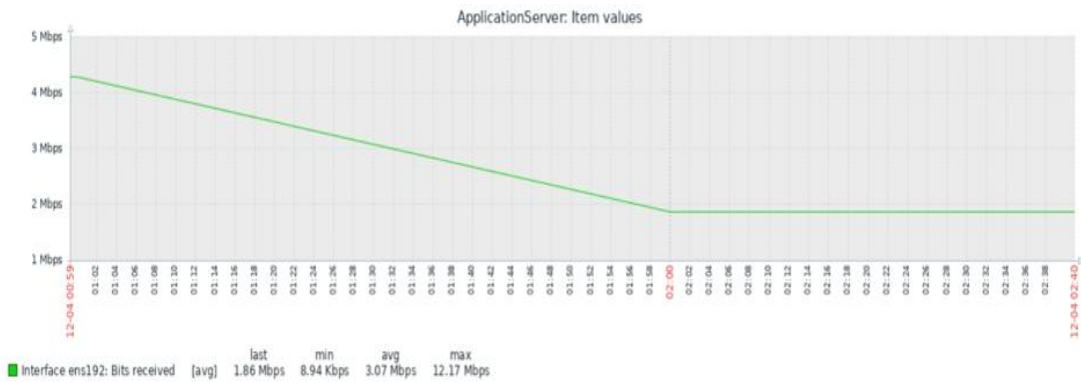
sistem tarafından yapılıyor ve alarm oluşuyor. Çalışan eforu da en aza indirgeniyor. 02.07 saatinde Interface sent değerinde %52 düşüş gözlemleniyor. 02.17 saatinde ise interface receive değerinde %63 düşüş gözlemleniyor. Son 4 haftanın baseline verileri baz alındığında son 15 dakikalık interface kullanım ortalamasında anormal bir düşüş olduğu, ilk alarmda sent verisi için yaklaşık yarı yarıya bir düşüş olmuşken ikinci alarmda %63 düşüş gözlemlenmiştir.

2024-12-04 02:17:07	Not classified	2024-12-04 02:22:07	RESOLVED	Applicat ionServ er	-ML- Interface Receive Kullanımında Anomali : Son 15 dakikalık Interface Receive ortalaması, Son 4 haftalık Interface Receive baseline verilerine göre %63 daha azdır. Son 15 Dk : 900044 Baseline Verisi : 2418367 ?
2024-12-05 02:07:09	Not classified	2024-12-05 02:12:09	RESOLVED	Applicat ionServ er	-ML- Interface Sent Kullanımında Anomali : Son 15 dakikalık Interface Sent ortalaması, Son 4 haftalık Interface Sent baseline verilerine göre %52 daha azdır. Son 15 Dk : 880883 Baseline Verisi : 1816816 ?

Şekil 5.17. Interface Düşük Kullanım Alarmları



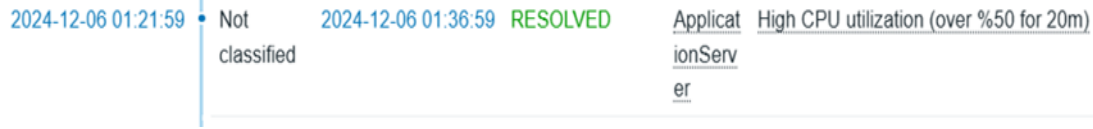
Şekil 5.18. Interface Sent Düşük Kullanım Grafiği



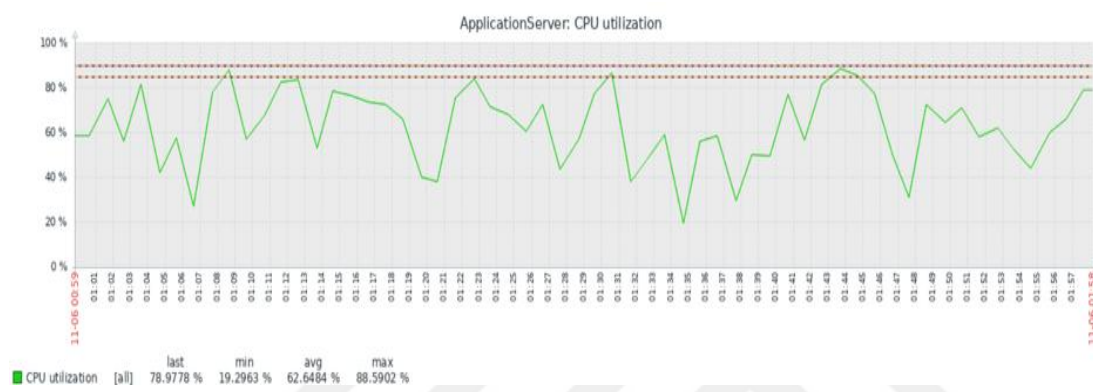
Şekil 5.19. Interface Receive Düşük Kullanım Grafiği

Bir başka senaryoda ise geleneksel izleme yönteminde alarm gelirken ML ile akıllı baseline oluşturulup durumun normal olduğu algılanmaktadır ve Şekil 5.21' deki grafik incelendiğinde ML alarmları yansımamıştır. Böylelikle haftanın her anı geçen

4 haftanın aynı anları ile değerlendirilmiş olup anomali olup olmadığı tespit edilmektedir. Geleneksel yöntemle ise yüksek cpu kullanım alarmı Şekil 5.20’deki gibi yansımıştır. Bu neticede makine öğrenmesi ile false-positive alarmlardan ve alarm sayısının da fazla olmasını bu akıllı yapı sayesinde en aza indirgenmiş oldu.

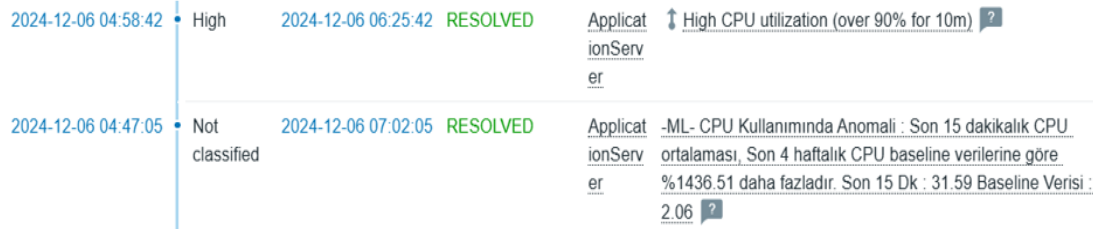


Şekil 5.20. CPU Yüksek Kullanım Alarmları

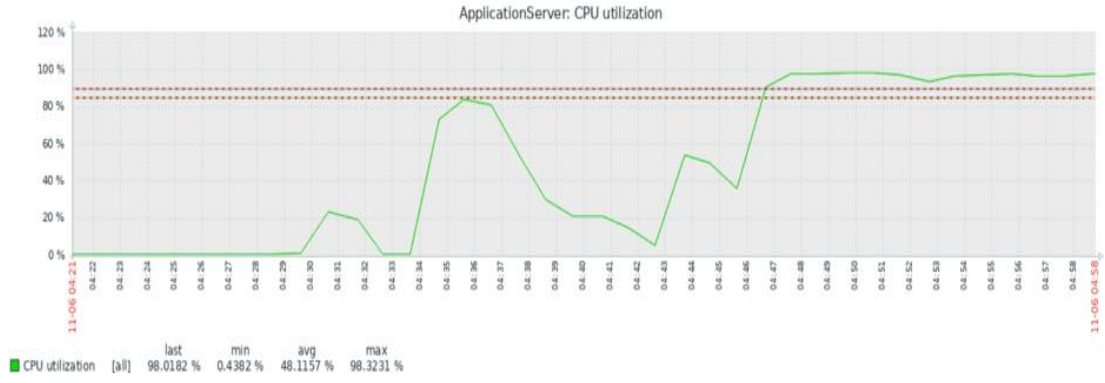


Şekil 5.21. CPU Yüksek Kullanım Grafiği

Bir diğer senaryoda ise makine öğrenmesi sistemlerinin sorunu daha önceden algılaması değerlendirilecektir. Şekil 5.22’de bulunan alarmlar incelendiğinde makine öğrenmesi alarmı cpu da anomali oluşumunu baseline ile değerlendirdiği için o aralıkta anomali olduğunu daha erken tespit etmektedir. Geleneksel izleme yönteminde ise alarm yaklaşık 10 dakika sonra gelmiştir. Günümüzü dikkate aldığımızda 10 dakika içinde bu sistem neredeyse kullanılamaz hale gelebilmektedir. Şekil 5.23’te bulunan cpu kullanım grafiğine de bakılırsa olaylar oluşmadan veya başlarken tespit edilmesi sorunların büyümeden, ilerlemeden çözülmesine olanak sağlayacaktır.



Şekil 5.22. CPU Yüksek Kullanım Alarmları

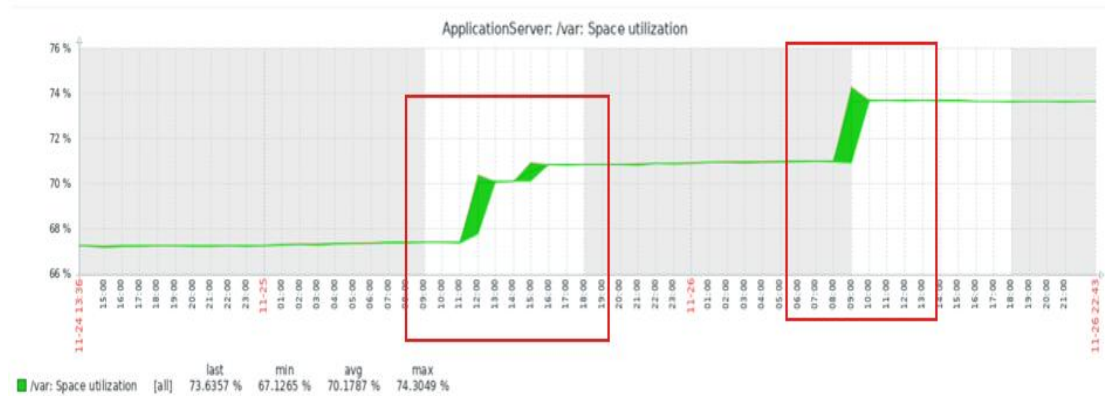


Şekil 5.23. CPU Yüksek Kullanım Grafiği

Bir diğer senaryoda ise Sistem disklerinin doluluğunun daha önceden tespiti ve ani artışı izlenmektedir. Bu da yine geleneksel izleme dışında sorun oluşmadan sistemlerin izlenmesini sağlamaktadır. Özellikle disk artışının ve eşik değerini ne zaman geçebileceğini ölçümlemesi büyük yapılardaki yoğun kullanılan önemli diskler için büyük önem taşımaktadır. Sunucumuzda bulunan /var dizininin bu şekilde kullanımı devam eder ise 7 gün içerisinde %90 eşik değerini aşacağını belirten bir akıllı alarm Şekil 5.24’ te bulunmaktadır. Alarm 25 kasımda düzelirken 26 kasımda tekrar bir artış tespit edilince ikinci kez yansımıştır. Artışlar Şekil 5.25’ te grafiksel olarak görülmektedir. Ekiplerin önceden altyapı hazırlığı yapmasına, kendilerini ayarlama ve iş planı yapmasına büyük katkı sağlamaktadır.

2024-11-26 16:47:01	• Average	2024-11-26 23:57:01	RESOLVED	Applicati onServe r	--ANOMALY-- /var: Disk kapasitesi 7 gün içerisinde (used > 90%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.) ?
2024-11-25 19:17:01	• Average	2024-11-26 07:21:01	RESOLVED	Applicati onServe r	--ANOMALY-- /var: Disk kapasitesi 7 gün içerisinde (used > 90%) eşik değerini aşacaktır. (Daha hızlı artış olur ise beklenenden erken dolum olacaktır.) ?

Şekil 5.24. Disk Kullanımı Artış Alarmları

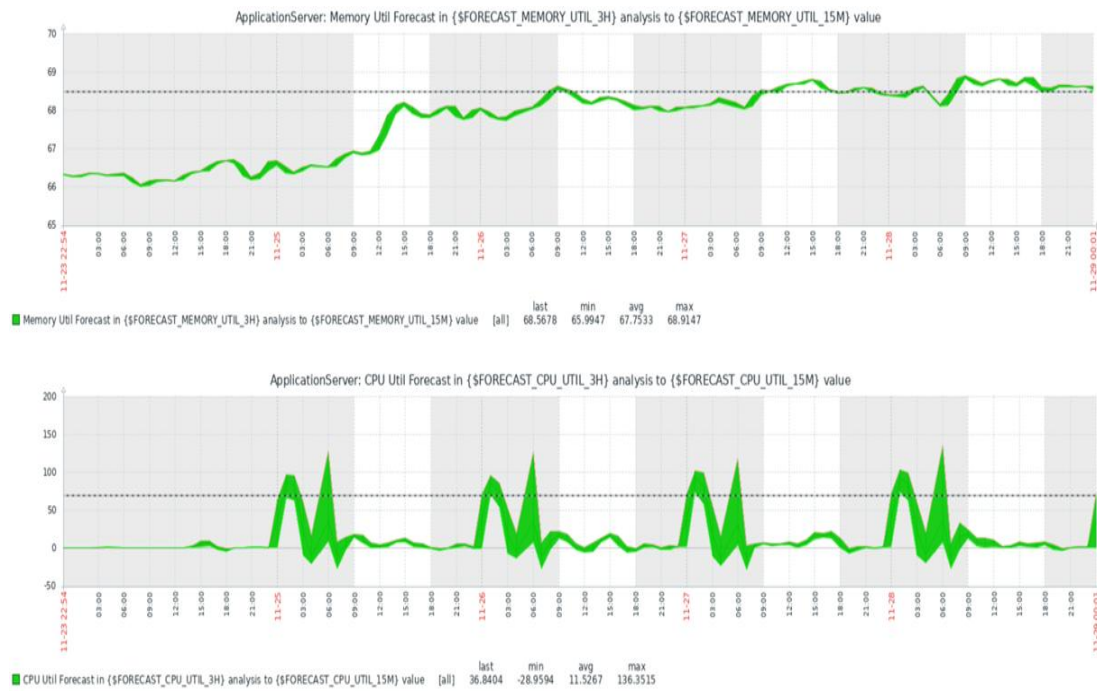


Şekil 5.25. Disk Kullanımı Artış Grafiği

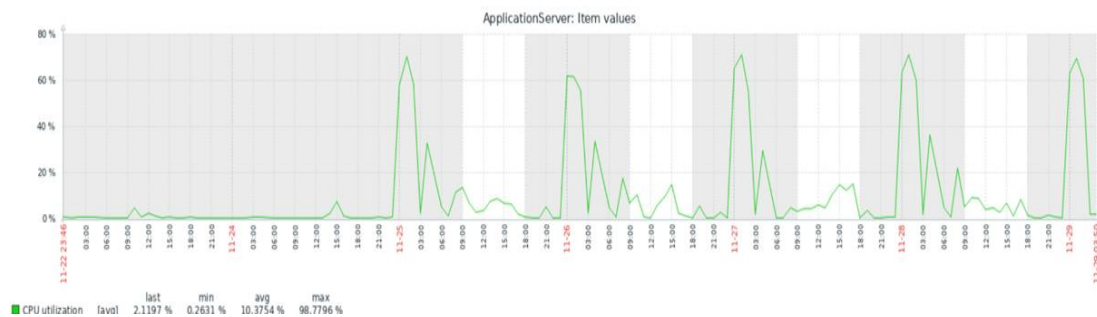
Bir başka senaryoda ise sunucu üzerinde memory ve cpu kullanımlarının geçmişe göre değerlendirilerek gelecekte ne kadar olabileceğinin tahminlemesi yapılmıştır. Şekil 5.26’ da gelen alarmlar baz alınarak sistemin cpu ve memory değerlendirmeleri Şekil 5.27’ de eklenmiştir. Burada Şekil 5.29’ da bulunan cpu kullanımına bakıldığında aslında tahminlemenin yaklaşık olarak paralel ilerlediğini yükselme yaptığı noktalarda yükseldiğini, düşüş anında da yine düşüş yaşadığı görülmektedir.

2024-11-30 00:52:53	Not classified	2024-11-30 01:02:53	RESOLVED	ApplicationServer	Memory Util Forecast in 3h analysis to 15m value
2024-11-29 06:25:54	Not classified	2024-11-29 06:29:54	RESOLVED	ApplicationServer	CPU Util Forecast in 3h analysis to 15m value

Şekil 5.26. Tahminleme Alarmları



Şekil 5.27. Cpu ve Memory Kullanımının Gelecekteki Tahminlemesi



Şekil 5.28. CPU Kullanımının Gelecekteki Tahminlemesi ile Karşılaştırılması

Bu incelemeler sonucunda, geleneksel izleme sistemleri ile makine öğrenmesi tabanlı izleme sistemlerinin karşılaştırmalı analizi gerçekleştirilmiş ve veri merkezi

operasyonlarında makine öğrenmesi yaklaşımlarının sağladığı detaylar ortaya konulmuştur. Performans karşılaştırmasını değerlendirdiğimiz de geleneksel izleme sistemleri ile makine öğrenmesi tabanlı izleme sistemleri arasında makine öğrenmesi yaklaşımlarının önemli üstünlükleri tespit edilmiştir. Makine öğrenmesi algoritmaları, geleneksel sistemlere kıyasla yüksek doğrulukla anomalileri tespit edebilmiştir. Alarm gelmeyecek noktalar anomaliyi algıladığı her durumda baseline verilerini de baz alarak alarm üretmiştir. ML tabanlı sistemlerin baseline analizi ile geleneksel sistemlere oranla yanlış pozitif alarm sayısında ciddi oranda düşüş görülmektedir. Normalde belirli eşik değerleri ile alarm gelirken makine öğrenmesi tanımlardan alarm yansımamıştır.

Proaktif izleme sistemleri, potansiyel sorunları ortalama 2-3 saat öncesinden tespit edebilmektedir. Özellikle ani yükselişlerde 10 dakia önceden sorun boy göstermeye başlamadan alarmlar yansımıştır. Disklerde oluşabilecek dolulukları ise 7 gün önceden tespit edebilmektedir. Lineer artışlarda daha da önceden tespit yapabilmesi mümkündür. Makine öğrenmesi tabanlı izleme sistemlerinin ekonomik etkileri düşünüldüğünde tahmine dayalı bakım stratejileri sayesinde sistem bakım maliyetlerinde tasarruf sağlanması muhtemeldir. Burada disklerin ne zaman dolacağı tahmini önceden hazırlıklara da imkân sunmuştur proaktif izleme mekanizmaları, sistem kesinti sürelerine olumlu yansımadır. Sorunlara oluşurken veya oluşmaya başlarken müdahale edilmesi kesintinin önüne geçebilmektedir.

Zabbix üzerinde gerçekleştirilen makine öğrenmesi entegrasyonu, izleme sistemlerinin yeteneklerini önemli ölçüde genişletmiştir. Dinamik değerler ile kendi kendini değerlendirebilmesi sorunları tespit edebilmesi birçok default tanımın yerine geçmektedir. Haftanın her anı için geçmiş 4 haftanın verileri değerlendirilip, gerçek anomaliler tespit edilmiştir. Özellikle baseline oluşturularak ve ağırlıklı hareketli ortalama algoritmasının (WMA) kullanılması daha da doğru tespit sağladı. Bu yapı ile sistem disklerindeki doluluk artışları önceden tahmin edilerek proaktif müdahaleler gerçekleştirilmektedir. Disk doluluk tahminleri büyük veri ve önemli ortamlarda, örneğin; e-posta sistemleri veri transfer sistemlerinde büyük öneme sahip. Son olarak geleneksel eşik değerlerinin ötesinde, makine öğrenmesi algoritmalarıyla daha akıllı alarm yönetimi sağlanmıştır. Sürekli tetiklenen alarm mekanizmaları değil akıllı yöntemler kullanılmıştır.

Tablo 5.3. Geleneksel İzleme ile Makine Öğrenmesi İzlemesinin Karşılaştırılması

Özellik	Geleneksel İzleme	Makine Öğrenmesi ile İzleme
Yaklaşım	Statik kurallar ve manuel yapılandırmalar kullanılır.	Dinamik öğrenme algoritmaları ve otomatik ayarlamalar kullanılır.
Veri Analizi	Temel eşik belirleme ve olay bazlı izleme yapılır.	Anomali tespiti, tahminsel analiz ve trend tahmini yapılır.
Ölçeklenebilirlik	Büyük veri kümelerini işlemek zordur; manuel müdahale gerektirir.	Büyük veri kümelerinde verimli analiz yapılabilir; ölçeklenebilir.
Sorun Tespiti	Sadece önceden tanımlanmış sorunları tespit edebilir.	Beklenmeyen veya bilinmeyen sorunları da tespit edebilir.
Reaksiyon Süresi	Sorunlar genellikle geç fark edilir; manuel müdahale gereklidir.	Sorunları gerçek zamanlı olarak tespit edip hızlı bir şekilde çözüm önerileri sunar.
Tahmin Yeteneği	Gelecekte oluşabilecek sorunları tahmin edemez.	Geçmiş verilerden öğrenerek potansiyel sorunları tahmin eder.
Karmaşıklık Seviyesi	Basit sistemlerde etkilidir ancak karmaşık sistemlerde yetersiz kalır.	Karmaşık ve dinamik sistemlerde etkili analiz ve izleme sağlar.
Manuel Müdahale	Sürekli manuel müdahale ve yapılandırma gerektirir.	Minimal manuel müdahale ile kendi kendine öğrenebilir ve ayarlanabilir.
Kullanım Alanları	Geleneksel ağ izleme, basit uygulama ve altyapı izleme.	Veri merkezleri, IoT, mikro hizmet mimarileri ve dinamik sistemlerin izlenmesi.
Avantajlar	Basit, düşük maliyetli ve anlaşılması kolaydır.	Daha derin içgörüler sağlar, geleceğe yönelik aksiyonlar almayı mümkün kılar.
Dezavantajlar	Karmaşık ve büyük sistemlerde yetersizdir; sürekli güncelleme gerektirir.	İlk kurulum ve model eğitimi daha karmaşıktır; daha fazla hesaplama gücü gerektirir.

Tablo 5.3 incelendiğinde, geleneksel izleme yöntemleri ile makine öğrenmesi tabanlı izleme sistemleri arasında, makine öğrenmesi tabanlı yaklaşımların belirgin bir üstünlük sağladığı açıkça görülmektedir.

Özellikle dikkat çeken noktalar arasında, makine öğrenmesi tabanlı sistemlerin proaktif yapısıyla sorunları önceden tahmin edebilmesi, süreçlerin daha hızlı ve verimli bir şekilde yönetilmesini sağlaması ve otomatize yöntemler sayesinde insan müdahalesine olan ihtiyacı önemli ölçüde azaltarak iş gücü tasarrufu sunması yer almaktadır. Bunun yanı sıra, sürekli öğrenme ve adaptasyon yeteneği sayesinde değişen koşullara hızlı bir şekilde uyum sağlayabilmesi, bu teknolojinin avantajlarını daha da belirgin hale getirmektedir.

Tablo 5.4. Geleneksel İzleme ile Makine Öğrenmesi Analizleri

Metod	Geleneksel İzleme	Makine Öğrenmesi ile İzleme
Alarm Sayısı	150-200	5-8
False / Positive Alarm Sayısı	150	YOK
Düşüş Eğilim Ölçümleri	Kısıtlı	Daha gelişmiş şekilde
Geleceğe Yönelik Alarm	YOK	VAR
Veri Zamansal Ağırlığı	YOK	VAR
Tahminleme	Gelecekte oluşabilecek sorunları tahmin edemez.	Geçmiş verilerden öğrenerek potansiyel sorunları tahmin eder.
İnsan Eforu	ÇOK	AZ
Dezavantajlar	Karmaşık ve büyük sistemlerde yetersiz kalmakta sürekli insan eforu gereklidir.	İlk kurulum ve model eğitimi daha karmaşıktır; daha fazla hesaplama gücü gerektirir. Kurulumu ve optimizasyonu zaman alır.

Tablo 5.4’ te görüleceđi üzere çıktılarımız sonucunda geleneksel izlemenin daha dar ve kısıtlı imkanlara sahip olduđunu ve insan eforuna çođunlukla ihtiyaç duyduđu görölmüştür. Makine öğrenmesi ile izlemelerde ise tam tersi daha gelişmiş şekilde geçmişten öğrenerek geleceđe yönelik ışıık tutması ve sorunları önceden tespit edebilmesi önem arz etmektedir. İnsan eforuna da proaktif olması nedeniyle daha az ihtiyaç duymakta ve yapıyı dinamikleştirmektedir.



6. SONUÇLAR VE TARTIŞMA

Veri merkezlerinde makine öğrenmesi tabanlı proaktif izleme sistemlerinin uygulanması, geleneksel reaktif izleme yaklaşımlarına kıyasla sistemlerin performans ve güvenilirliğini önemli ölçüde artıran devrim niteliğinde bir yaklaşımdır. Bu gelişmiş izleme mekanizmaları, karmaşık donanım ve yazılım altyapılarındaki potansiyel sorunları gerçek zamanlı olarak analiz ederek, olası arızaları ve performans dar boğazlarını önceden tespit etme yeteneğine sahiptir. Bu tespitler sonrası veri merkezi yöneticilerine, sistemin mevcut durumu ve gelecekteki olası senaryolar hakkında gerçek zamanlı, veri güdümlü karar destek mekanizmaları sağlayarak operasyonel verimliliği üst düzeye çıkarır. Makine öğrenmesi algoritmalarının gelişmiş analitik yetenekleri, veri merkezi altyapısındaki performans eğilimlerini ve kaynak tüketim desenlerini derinlemesine analiz eder. Sistemlerin güvenilirliğini artıran bu teknoloji, veri merkezi yönetiminde öngörücü ve optimize edilmiş bir yaklaşımın benimsenmesine öncülük eder. İleri düzey makine öğrenmesi modellerinin entegrasyonu, veri merkezi operasyonlarında daha esnek, dayanıklı ve verimli bir ekosistem oluşturur. Bu sistemler, karmaşık altyapı bileşenlerinin davranışlarını sürekli öğrenerek ve analiz ederek, geleneksel yönetim yaklaşımlarının ötesine geçen bir işletimsel mükemmellik sunmaktadır.

Makine öğrenmesi tabanlı proaktif izleme sistemleri, modern veri merkezlerinin performans, güvenilirlik ve maliyet etkinliği hedeflerine ulaşmasında kritik bir rol oynamaktadır. Makine öğrenmesi tabanlı izleme sistemleri, veri merkezlerinin sadece bugünkü ihtiyaçlarına yanıt vermekle kalmayıp, gelecekteki zorluklara karşı da hazır olmasını sağlayan kapsamlı bir çözüm sunmaktadır. Bu yenilikçi yaklaşımlar, veri merkezleri için daha sürdürülebilir, verimli ve güvenilir bir ekosistem yaratma potansiyeline sahiptir. Özellikle günümüzde artık hayatımızda yer edinmeye başlayan yapay zekâ bu proaktif izleme sistemleri içinde yapay zekâ destekli tahminsel bakım stratejileri, donanım bileşenlerinin ömrünü uzatır, beklenmedik kesintilerin önüne geçer. Bu alanda Yapay zekâ çalışmaları da büyük önem taşımaktadır.

Bu çalışmada, geleneksel izleme sistemleri ve makine öğrenmesi ile izleme sistemlerinin karşılaştırılması yapılmıştır. Açık kaynak bir izleme sistemi olan Zabbix'i tasarlayarak bu yapı üzerinde geleneksel izlemeleri konfigüre edildi. Bu tanımlar sistemi sürekli olarak izleyip sorun olması durumunda tanımlı veriler göre alarm oluşturur. Bu tanımlar yanında sistemimizde makine öğrenmesi desteği olan izleme tanımları da sisteme yerleştirilmiştir. Aynı ortamı her iki farklı tanımlar ile izleyerek sistemin bize oluşturduğu verilere çalışmamızda yer vererek avantajlarını, farklılık ve farkındalığı değerlendirildi. Bu çıktıların sonuçlarına göre geleneksel izlemelerde çok fazla alarm gelmesi veya tam tersi anomali oluştuğunda bildirim gelmemesi bir dezavantaj oluşturmaktadır. Ayrıca reaktif olarak sabit değerlere göre izleme olduğu için her izlenen cihaz için ayrı ayrı konfigürasyonlara yani dışardan müdahalelere ihtiyaç duymakta bu da eforu artırmaktadır. Makine öğrenmesi yöntemleri ile özellikle proaktif şekilde izlenmesi insan eforunu azaltmakta otomatik olarak kendi baseline ını oluşturması buna göre her cihazda ayrı bir şekil alması ve kendi kendine iyileşmesi büyük bir avantaj sağlamakla birlikte anomalilerin de yakalanmasını ve ilgili noktalara bildirilmesini sağlamıştır. Bu yüzden Makine öğrenmesi yöntemleri izleme sistemlerinin kaçınılmaz bir noktası olacağı bu çalışmada da öne çıkmıştır.

Araştırma sonuçlarına göre gelecek çalışmalar için önerilerden bahsedilecek olursa; veri merkezi izleme sistemleri için, daha gelişmiş makine öğrenmesi modellerinin entegrasyonu, gerçek zamanlı analiz yeteneklerinin daha da artırılması ve piyasa da bulunan ürünlere entegrasyon desteklerinin sağlanması ve bulut ve dağıtık sistemler için ölçeklenebilir izleme çözümlerinin geliştirilmesi, bu yönteminde proaktif tabana oturtulması üzerine olacaktır. Bu çalışmaların nihai uç noktası ise önce yapay zekâ sistemlerinin bu bahsettiğimiz çalışmaların en tepesinde yer alması ve AIOPS (BT için Yapay Zekâ Operasyonları)' a giden ve şekil alan bir tablo oluşmaktadır.

KAYNAKLAR

- Abualsauod, E. H. (2023). Machine learning based fault detection approach to enhance quality control in smart manufacturing. *Production Planning & Control*, 1-9. <https://doi.org/10.1080/09537287.2023.2175736>
- Ahuja, A. S. (2019). The impact of artificial intelligence in medicine on the future role of the physician. *PeerJ*, 7, e7702. <https://pubmed.ncbi.nlm.nih.gov/31592346/>
- Akgül, I. (1994). Zaman serisi analizi ve öngörü modelleri. *Öneri Dergisi*, 1(1), 52-69. <https://dergipark.org.tr/tr/download/article-file/992756>
- Alam, M., Islam, M. R., & Shil, S. K. (2023). AI-Based predictive maintenance for US manufacturing: reducing downtime and increasing productivity. *International Journal of Advanced Engineering Technologies and Innovations*, 1(01), 541-567.
- Andreolini, M., Colajanni, M., & Pietri, M. (2012, December). A scalable architecture for real-time monitoring of large information systems. In *2012 Second Symposium on Network Cloud Computing and Applications* (pp. 143-150). IEEE. <https://doi.org/10.1109/NCCA.2012.24>
- Netscout Documentation (2024). Artificial Intelligence (AI) & Machine Learning (ML) Usage in Arbor Edge Defense, (2024). Erişim Zamanı 25.11.2024, <https://www.netscout.com/resources/quick-looks/artificial-intelligence-ai-machine-learning-ml-usage-in-arbor-edge-defense>
- Arslan İ. (2024) What is Big Data? Erişim Zamanı 25.11.2024, <https://www.trexakademi.com/tr/dijital-arac/big-data-nedir-8>
- Berus, Y., & Yakut, Y. B. (2024). Derin Öğrenme (1D-CNN, RNN, LSTM, BiLSTM) ile Enerji Tüketim Tahmini: Diyarbakır AVM Örneği. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 15(2), 311-322. <https://doi.org/10.24012/dumf.1415055>
- Birje, M. N., & Bulla, C. (2019). Cloud monitoring system: basics, phases and challenges. *International Journal of Recent Technology Engineering (IJRTE)*, 8(3), 4732-4746. <http://www.doi.org/10.35940/ijrte.C6857.098319>
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 1-58.
- Chen M., What Is Machine Learning? Erişim Zamanı 27.11.2024, <https://www.oracle.com/tr/artificial-intelligence/machine-learning/what-is-machine-learning/>
- Datadog Documentation, (2024). Erişim Zamanı 02.12.2024, <https://docs.datadoghq.com/>

- Dong, H., Munir, A., Tout, H., & Ganjali, Y. (2021). Next-generation data center network enabled by machine learning: Review, challenges, and opportunities. *IEEE Access*, 9, 136459-136475. <https://doi.org/10.1109/ACCESS.2021.3117763>
- Everett, M., & Thomas, G. (2024). Machine Learning-Powered Dynamic Resource Allocation for Sustainable Cloud Infrastructure. <http://dx.doi.org/10.13140/RG.2.2.23402.73924>
- Evonsys. (n.d.). Effectively Using Zabbix Tool for Network Monitoring. Erişim Zamanı 25.11.2024, <https://www.evonsys.com/blog/effectively-using-zabbix-tool-for-network-monitoring>
- Eyþórsson, G. G. (2017). Improving Monitoring of IT Systems using Machine Learning (Doctoral dissertation).
- Farbiz, F., Miaolong, Y., & Yu, Z. (2020, November). A cognitive analytics based approach for machine health monitoring, anomaly detection, and predictive maintenance. In 2020 15th IEEE conference on industrial electronics and applications (ICIEA) (pp. 1104-1109). IEEE. <https://doi.org/10.1109/ICIEA48937.2020.9248409>
- Goswami, M. J. (2024). AI-Based Anomaly Detection for Real-Time Cybersecurity. *International Journal of Research and Review Techniques*, 3(1), 45-53.
- Günel, A. (2014). *Üniversitelere yönelik yeni bir veri merkezi tasarımı ve uygulaması* (Master's thesis, Bilecik Şeyh Edebali Üniversitesi, Fen Bilimleri Enstitüsü). <https://acikkaynak.bilecik.edu.tr/xmlui/bitstream/handle/11552/219/10038962.pdf?sequence=1&isAllowed=y>
- Jani, Y. (2023). AI and Machine Learning for Predictive Monitoring in IT Operations. <http://dx.doi.org/10.21275/SR24822171836>
- Kavis, M. (2014). *Architecting the cloud: design decisions for cloud computing service models (SaaS, PaaS, and IaaS)*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Langley, P., & Simon, H. A. (1995). Applications of machine learning and rule induction. *Communications of the ACM*, 38(11), 54-64. <https://doi.org/10.1145/219717.219768>
- Li, S., Yin, Q., Li, G., Li, Q., Liu, Z., & Zhu, J. (2022, June). Unsupervised contextual anomaly detection for database systems. In *Proceedings of the 2022 International Conference on Management of Data* (pp. 788-802).
- Nagios Documentation (2024). Erişim Zamanı 29.11.2024, <https://www.nagios.org/documentation/>
- Nardelli, P., Papadias, C., Kalalas, C., Alves, H., Christou, I. T., Macaluso, I., ... & Alonso-Zarate, J. (2019, August). Framework for the identification of rare events via machine learning and IoT networks. In 2019 16th International symposium on wireless communication systems (ISWCS) (pp. 656-660). IEEE. <https://doi.org/10.1109/ISWCS.2019.8877287>
- New Relic Documentation (2024). Comprehensive Application and Infrastructure Monitoring with AI-Powered Insights. Erişim Zamanı 25.11.2024, <https://docs.newrelic.com/>

- NR, M. K., Nijaguna, G. S., Chaithra, K. N., Alsalami, Z., & Kumari, K. A. (2024, March). eMRA: A Multi-Objective Resource Allocation Technique using Genetic Algorithms for Cloud Computing. In 2024 International Conference on Distributed Computing and Optimization Techniques (ICDCOT) (pp. 1-5). IEEE. <https://doi.org/10.1109/ICDCOT61034.2024.10516192>
- Olaoye, F., & Potter, K. AI-Driven Anomaly Detection in Critical Infrastructure. Erişim Zamani 25.11.2024, https://www.researchgate.net/publication/383565282_AI-Driven_Anomaly_Detection_in_Critical_Infrastructure
- Prometheus Documentation (2024). Prometheus documentation. Erişim Zamani 25.11.2024, <https://prometheus.io/docs>
- Robles-Durazno, A., Moradpoor, N., McWhinnie, J., & Russell, G. (2018, June). A supervised energy monitoring-based machine learning approach for anomaly detection in a clean water supply system. In 2018 international conference on cyber security and protection of digital services (Cyber Security) (pp. 1-8). IEEE. <https://doi.org/10.1109/CyberSecPODS.2018.8560683>
- Saini, N., Yadav, A. L., & Rahman, A. (2024, March). Cloud Based Predictive Maintenance System. In 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 1-5). IEEE. <https://dx.doi.org/10.1109/ICRITO61523.2024.10522398>
- Shabbir A., Arshad N., Rahman S., Md Sayem A. & Chowdhury F. (2024). Shabbir, A., Arshad, N., Rahman, S., Sayem, M. A., & Chowdhury, F. (2024). Analyzing surveillance videos in real-time using AI-powered deep learning techniques. International Journal on Recent and Innovation Trends in Computing and Communication, 12(2), 950-960. https://www.researchgate.net/publication/384459473_Analyzing_Surveillance_Videos_in_Real-Time_using_AI-Powered_Deep_Learning_Techniques
- Mohamed, S., Sarr F., (2024). The Role of Predictive Monitoring in Transforming IT Operations: A Framework for Early Detection and Prevention of System Failures. <https://journalgate.com/index.php/IJDI/article/view/17>
- Thiyagarajan, K., Kodagoda, S., & Van Nguyen, L. (2017, June). Predictive analytics for detecting sensor failure using autoregressive integrated moving average model. In 2017 12th IEEE conference on industrial electronics and applications (ICIEA) (pp. 1926-1931). IEEE. <https://doi.org/10.1109/ICIEA.2017.8283153>
- Toka, L., Dobreff, G., Haja, D., & Szalay, M. (2021, May). Predicting cloud-native application failures based on monitoring data of cloud infrastructure. In 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM) (pp. 842-847). IEEE.
- Yu, B., Zhang, Y., Xie, W., Zuo, W., Zhao, Y., & Wei, Y. (2023). A network traffic anomaly detection method based on Gaussian mixture model. Electronics, 12(6), 1397. <https://dx.doi.org/10.3390/electronics12061397>
- Zabbix Documentation (2024). Zabbix documentation. Erişim Zamani 25.11.2024, <https://www.zabbix.com/documentation>

Zabbix Zirvesi (2021). Machine Learning in Zabbix 6.0 LTS: Anomaly Detection and Baselines. Eriřim Zamanı 1.11.2024, https://assets.zabbix.com/files/events/zabbix_summit_2021/slides/Alex_Kalinulin_Machine_learning_baseline.pdf/

Zhang M., Next Generation Data Center: Insights into the Next-Gen Future. Eriřim Zamanı 25.11.2024, <https://dgtlinfra.com/next-generation-data-center/>



ÖZGEÇMİŞ

Ad-Soyad : Metehan HAN

ÖĞRENİM DURUMU:

- **Lisans** : 2019, Atatürk Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Önlisans** : 2018, Atatürk Üniversitesi, Açıköğretim Fakültesi, İş Sağlığı ve Güvenliği
- **Önlisans** : 2021, Atatürk Üniversitesi, Açıköğretim Fakültesi, Adalet

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2019 yılı 2242 Üniversiteler Arası TÜBİTAK Birinciliği
- 2020-2021 yılları arasında Kuveyt Türk Katılım Bankası'nda Otomasyon ve Monitoring Sistemleri alanında olarak çalıştı.
- 2021-2023 yılları arasında Türk Telekom AssisTT' te Monitoring Sistemleri Admini olarak çalıştı.
- 2023 – yılı itibariyle Sigorta Bilgi ve Gözetim Merkezi'nde Monitoring Sistemleri Admini olarak çalışıyor.
- 2024 yılı Future of AI & Cloud Zirvesinde Makine Öğrenmesi Kullanarak Proaktif Anomali Yönetimi Projesi Ödülü

TEZDEN TÜRETİLEN ESERLER:

- Sevin, A. Ve Han, M. B. 2024. Machine Learning Based Proactive Monitoring System In Data Centers, *International 6th Bursa Scientific Researches Congress*. ISBN: 978-625-6997-55-4, 962-981.