# Sistem Yönetimi Hizmetleri için Sorun Tahmini ve Önleyici Bakım Programı Geliştirilmesi

Sertay Özer[1], Razık Harsoğlu[1], Erdem Akdoğan[1], Yeliz Ekinci[2], Engin Zorlu[1,]

[1] Uzman Bilişim AR-GE Merkezi, Kartal, İstanbul
[2] Bilgi Üniversitesi, Eyüp, İstanbul
[1]{sertay.ozer,razık.harsoglu, erdem.akdogan, engin.zorlu}@experteam.com.tr
[2]yeliz.ekinci@bilgi.edu.tr

**Özet.** İlişkisel veritabanı yönetim sistemleri (İVYS) bankacılık, telekom ve benzeri birçok alandaki kritik yazılımların işleyişinde önemli bir rol oynamaktadır. Bu sistemlerde oluşan hatalar ve kesintilerin büyük maliyetleri olabilmektedir. Bu sebeple bu sistemlerin sürekli takibi ve hataların önceden tahmin edilebilmesi büyük önem taşımaktadır. Mevcut durumda sistemlerde oluşan teknik hatalar uzman personel tarafından takip edilmekte ve sorun oluştuktan sonra personel tarafından çözüm üretilmesi beklenmektedir. Bu da verimlilik ve zaman kaybına neden olmaktadır. Sorunlar genellikle kritik hale geldiğinde ortaya çıktığı için hem çözümün uygulanması zorlaşmakta hem de kullanıcıların iş süreçleri sorundan etkilenmektedir. Bu çalışmada belirtilen yöntemlerle veritabanı ve orta katman ürünlerini kullanan firmalardaki sistem altyapılarında meydana gelen teknik problemlerin, ortaya çıkmadan önce tahmin edilerek, düzeltici aksiyonların önerilmesini sağlayacak makine öğrenmesi tabanlı bir erken uyarı sistemi geliştirilmiştir. Bu sistem farklı kurumlarda çalışan İVYS'lerden saatlik olarak kütük kayıtlarını toplamakta ve toplanan kayıtlar önişleme tabi tutularak karar ağacı tabanlı makine öğrenmesi modeline beslenmektedir. Yapılan sınamalarda 5694 noktalık, çoğunluğu kritik hata kayıtlarından oluşan veri kümesinde karar ağacı algoritmalarının %98 doğrulukla çalıştığı gözlemlenmiştir.

**Keywords:** Veritabanı yönetimi, Tahminlenme, Makine öğrenmesi, Veri madenciliği

# Problem Estimation and Preventive Maintenance Program Development for System Management Services

**Summary.** Relational database management systems (RDBMS) play an important role in the operation of critical software in many areas such as banking and telecom. In these systems, faults and interruptions can be very costly. For this reason, it is very important for these systems to be able to track systems continuously for faults and predict faults before they happen. In the present situation, the technical faults that occur in the systems are followed by the expert personnel and it is expected that the personnel will produce a solution after the problem occurs. This leads to productivity and time loss. Problems often become more difficult to solve, as the problems often arise when they become critical, and the business processes of users are affected. In this study, an early warning system based on machine learning has been developed to predict RDBMS faults before the emergence of technical problems in the system infrastructures of firms using database and middleware products. This system collects the log records on the hourly basis from the RDBMS

working in different institutions and is fed to the decision tree based machine learning model after preprocessing operations. It has been observed that the decision tree algorithms in the dataset, which consist of 5694 points and a majority of the critical error registers, run with 98% accuracy.

**Keywords:** Database Management, Prediction, Machine learning, Data mining

## 1 Giriş

İlişkisel Veritabanı Yönetim Sistemleri (İVYS) büyük miktarlardaki ilişkisel ve yapısal verinin bir çok güvenlik mekanizması ile korunarak saklandığı ve erişimin sağlandığı, çok sayıda kullanıcıya aynı anda yazma okuma güncelleme ve silme erişimi imkanlarının sağlandığı programlardır. Veritabanı ve orta katman ürünleri kullanan büyük ve orta ölçekli firmalarda çalışan sistem altyapılarında, çalışma kalitesini düşüren teknik problemler ortaya çıkabilmektedir. Bu teknik problemler ortaya çıkmadan önce tahmin edilerek sınıflandırılması ve buna bağlı olarak doğru düzenleyici eylemlerin önerilebilmesi sistemlerin kesintiye uğramadan, verimli bir şekilde çalışmasına olanak sağlayabilir.

Uzman Bilişim A.Ş. (ExperTeam) müşterilerine sahip olduğu bilgi işlem ortamlarının sağlıklı ve kesintisiz bir şekilde işlemesini sağlamak için "Sistem Yönetimi" hizmeti kapsamında özellikle İVYS alanında danışmanlık, bakım, sorun giderme, kurtarma hizmetleri sağlamaktadır. Bu sistemlerde oluşan hatalar ve kesintilerin büyük maliyetleri olabilmektedir. Bu sebeple bu sistemlerin sürekli takibi ve hataların önceden tahmin edilebilmesi büyük önem taşımaktadır. Mevcut durumda sistemlerde oluşan teknik hatalar uzman personel tarafından takip edilmekte ve sorun oluştuktan sonra personel tarafından çözüm üretilmesi beklenmektedir. Bu da verimlilik ve zaman kaybına neden olmaktadır. Sorunlar genellikle kritik hale geldiğinde ortaya çıktığı için hem çözümün uygulanması zorlaşmakta hem de kullanıcıların iş süreçleri sorundan etkilenmektedir.

Bu çalışmada belirtilen yöntemlerle veritabanı ve orta katman ürünlerini kullanan firmalardaki sistem altyapılarında meydana gelen teknik problemlerin, ortaya çıkmadan önce tahmin edilerek düzeltici eylemlerin önerilmesini sağlayacak makine öğrenmesi tabanlı bir erken uyarı sistemi geliştirilmiştir. Bu sistem farklı kurumlarda çalışan İVYS'lerden saatlik olarak kütük kayıtlarını toplamakta ve toplanan kayıtlar önişleme tabi tutularak karar ağacı tabanlı makine öğrenmesi algoritmasına beslenmektedir. Yapılan değerlendirmelerde 5694 noktalık, çoğunluğu kritik hata kayıtlarından oluşan eğitim veri kümesi ile eğitilen karar ağacı algoritmasının, 1896 noktalık sınama verisi ile değerlendirildiğinde %98 doğrulukla çalıştığı gözlemlenmiştir.

## 2 Literatür Taraması

Günümüzde "Sistem Yönetimi" altyapısında oluşan sorunların çözümü için literatürde yapılan çalışmalar, sistemdeki sorunların sınıflandırılarak, doğru sınıflandırma yapan tahminleme yöntemlerinin geliştirilmesini önermektedir. Sorun sınıfının belirlenmesi ile, soruna uygun

çözüm önerisinin sunulması mümkün olabilmektedir. Makine öğrenmesi algoritmalarının anormallik tespitinde kullanılması ile ilgili bir örnek olarak internet ağ alt yapısı trafiğinde anormallik tespitinin yapılması verilebilir [11]. Alonso ve diğerlerinin çalışmasında [3] bilgisayar sistemlerinde karşılaşılan beklenmedik servis kesilmeleri problemi ele alınmıştır. Lan ve diğerlerinin çalışmasında [1] geniş çaplı sistemlerde, sistemdeki anormallikleri tespit edecek otomatik mekanizma tasarlamak üzerine benzer bir çalışma yapmıştır. Benzer çalışmalarda bu kesilmeler yazılımların yaşlanması olgusuyla açıklanmakta ve hafızanın aşırı doluluğu, sonlandırılmamış işler, verilerin kirlenmesi gibi etkenlerin yaşlanmaya neden olduğu belirtilmektedir [2], [5]. Bu tür problemlerin çözümü için iki temel çözüm kategorisi literatürde işlenmiştir. Bunlardan ilki zaman temelli uygulamalardır ve sistemin önceden belirlenmiş, periyodik zaman aralıklarında düzenli olarak gençleştirilmesi şeklinde uygulanmaktadır. İkincisi ve son yıllarda özellikle ele alınan kategori ise sistemdeki bazı ölçütleri düzenli olarak takip ederek, serviste oluşabilecek herhangi bir kesintiyi gerçekleşmeden kısa bir süre önce tahmin edip bunun akabinde gençleştirme işleminin yapılmasını öngörmektedir. Bu ölçütlere örnek olarak CPU kullanımı, boş hafıza, ağ trafiği ve girdi/çıktı oranları verilebilir [3], [1].

Literatürde genellikle sınıflandırma yapan makine öğrenmesi/veri madenciliği teknikleri kullanılarak problem sınıfı hakkında tahmin modelleri geliştirilmiştir. Bu modellerde, soruna yol açan değişkenler bağımsız değişken, sorunun sınıfı ise bağımlı değişken olarak kullanılmıştır. Lan ve diğerlerinin çalışmalarında [1] 5 tipte hata/sorun için tahmin modeli geliştirmişlerdir. Deneyler hataların tek tek enjekte edildiği ve birden çok hatanın bir arada enjekte edildiği ortamlarda gerçekleştirilmiştir. Toplamda 19 tane bağımsız değişken belirlenmiş ve bu değişkenlerle ilgili veri toplanmıştır. Lan ve diğerlerinin çalışmalarında [1] iki adet başarı kriteri belirlemişlerdir; hassaslık ve belirginlik; ve bu kriterlere göre daha yüksek başarı gösteren sınıflandırma modelini sonuç modeli olarak seçmişlerdir. Stewart ve diğerlerinin çalışmalarında [4] EntomoModel adlı karar ağacı modeli tabanlı bir mekanizma geliştirerek, kök sorunu da ortaya koyan bir anomali tahmini çalışması yapmışlardır. Alonso ve diğerlerinin çalışmasında [3] önerilen sınıflandırma tahmini yöntemi, sistemi üç fazda değerlendirmektir. Yeşil alarmın saptandığı durumda sistemin doğru bir şekilde çalıştığı, turuncu alarmın saptandığı durumda sistemin olası bir kesintiye karşı uyarı vermeye başladığı, kırmızı alarmın saptandığı durumda ise sistemin tehlike altında olduğu çıkarımları yapılmakta ve buna göre gereken önlemler alınmaktadır.

Yapılan literatür taraması ve firmadaki sorun tipleri incelendiğinde sorun tahmini için karar ağaçlarının kullanılmasına karar verilmiştir [3], [6], [4], [7], [8]. Karar ağacı algoritmaları düğümlerin değişkenlerin olası değerlerini sorguladığı, yukarıdan aşağıya doğru dallandırılan ve böylece sistem davranışını öğrenmeye çalışan algoritmalardır.

Literatürdeki çalışmalar, veri setini deneme, doğrulama ve test olmak üzere üç parçaya ayırır ve geliştirilen model için bazı başarı kriterleri belirler. Ardından pek çok makine öğrenmesi/veri madenciliği tekniği uygulayarak başarı kriterlerine göre özellikle test kümesinde en başarılı olan (en doğru sınıflandırmayı yapan) yöntemi seçer [9], [10]. Değişken sayısının fazla olduğu durumlarda, Lasso düzenleme teknikleri, PCA (Temel Bileşen Analizi) ve ICA (Bağımsız Bileşen Analizi) ve benzeri yöntemler kullanılarak sistemde izlenmesi gereken değişken sayısının azaltılması ve böylece seçilen sınıflandırma metodunun performansının iyileştirilmesi de mümkün olmaktadır [3], [1]. Model başarısını ölçerken genellikle karışıklık matrisinden yararlanılır. Karışıklık matrisinden yararlanılarak makalelerde en çok kullanılan başarı kriterleri doğruluk, kesinlik ve anmadır [10].

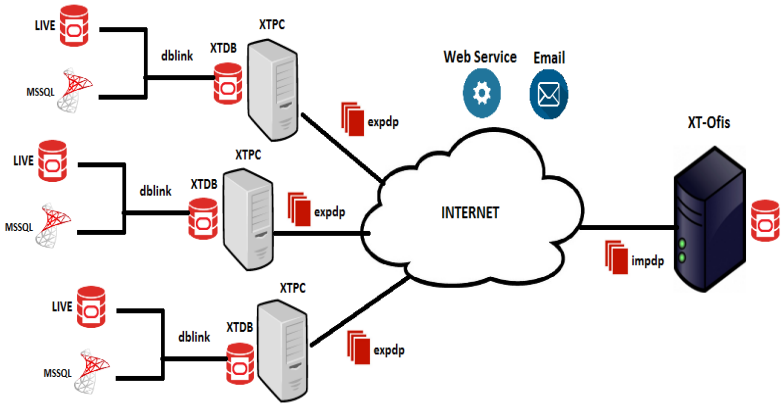
## 3 Yaklaşım

Sistem Yönetim Hizmetleri için Sorun Tahmini ve Önleyici Bakım Programı geliştirilmesinde aşağıda anlatılan yaklaşımlar izlenmiştir. Yönetimi yapılan sistemlerin belli ölçütlerinin izlenmesi, değerlendirilmesi ve gerekli aksiyonların alınması gerekmektedir. Oracle bazlı sistemlerden toplanan veriler 756 adet farklı metriği içermektedir. Örnek olarak şu metrikler verilebilir:

a. Geri Yükleme Hızı (KB/sec),
b. Boşta disk grubu (MB),
c. İşlemci kullanımı (saniye başına),
d. Yükleme İşlemci kullanımı (%),
e. Java Sanal Makinesi Hafıza Havuzu Kullanım Tavanı (KB)

Verilerin büyümesini engellemek amacıyla her seviyedeki veri için farklı tutulma süreleri bulunmaktadır. Bu veriler günlük özet şeklinde sürekli, saatlik özet olarak 2 ay boyunca tutulmaktadır. Dolayısı ile geçmişe ait veriler günlük özet olarak mevcuttur.

Şekil 1'de görülebileceği üzere izlenen İVYS sistemlerinden verilerinin toplanabilmesi için sistem üzerine kurduğumuz yazılım ile iki tip transfer metodu kullanılabilmektedir. E-posta ile ya da Webservis yöntemi ile izleme ölçütlerini içeren veri ofis ortamına gönderilebilmektedir. Güvenlik politikaları gereği izleme sunucularının internet adresine erişimi engellendiği durumlarda geliştirdiğimiz e-posta transferi yöntemi kullanılmaktadır.




**Şekil 1.** Oracle Sistemlerinden Kütük Verisi Toplama Mimarisi.

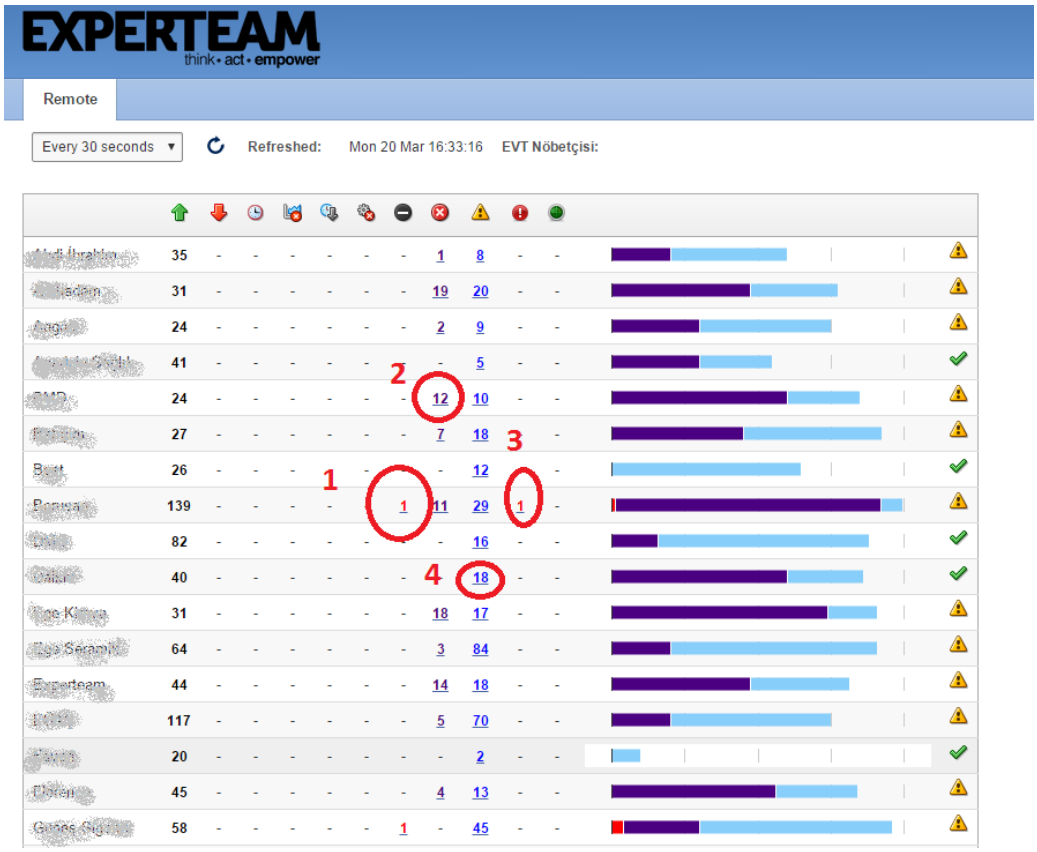
İlişkisel veritabanı üzerinde birden fazla tabloda tutulmakta olan metrik verilerinin analiz yapılabilecek formata çevrilmesi amacıyla bir veri akışı aracı ile okunmasına karar verildi. Burada kullanılacak veri akışı aracı seçimi yapılırken Flume, Storm, Spark gibi farklı opsiyonlar karşılaştırıldı ancak hem benchmark testlerinde daha iyi bir grafik gösterdiği hem de büyük

veri ile alakalı çalışmalarda giderek artmakta olan kullanım istatistiği nedeniyle Spark tercih edilmiştir. Spark tarafından desteklenen SQL benzeri yapı ile ilişkisel veri tabanı üzerinden veri çekmek kolay bir şekilde mümkün olduğundan öncelikle Oracle üzerinde birden fazla tablo üzerinde tutulmakta olan metrik verisi Spark bellek içi sisteme alınarak R tarafından üzerinde çalışılmak üzere anlaşılan veri yapısı formatına dönüştürüldü. Oracle üzerindeki tablolar devamlı yeni veri ile beslendiğinden ve Spark tarafında belirli aralıklarla çalışan bir program olacağından; son aktarılan kaydı belirten bir değeri tutma ihtiyacı doğdu. Bunu çözmek için Hadoop dağıtık dosya sistem üzerinde bir index dosyası oluşturuldu ve son okunan metriği gösteren değer aktarım sonrası buraya kaydedildi. Sonraki aktarımlarda da buradaki metrik değeri kontrol edilerek, bu index değerinden sonra aktarılmış olan metriklerin Spark'a aktarımı sağlanmıştır.

Üzerinde analiz yapılacak veri yapısı için zaman bağımlı olarak metriklerin aynı satırda toplanması ihtiyacı R ile kullanılacak metod nedeniyle ortaya çıktı ve ilişkisel veri tabanında satır olarak tutulan metrik verileri Spark üzerinde geliştirilen kod ile aynı satırda tutulacak hale getirildi. Üzerinde analiz yapılabilecek veri yapısına dönüştürülen metrik verileri yeni halleriyle Hadoop dağıtık disk sistemi üzerinde tutulmaktadır. Spark memory ve disk sistemini hibrid olarak kullanabildiğinden analiz işlemi sırasında veri boyutuna bakarak ve hafıza yeterliliğini kontrol ederek Hadoop dağıtık dosya sistemi de kullanabilmektedir.

Hadoop dağıtık dosya sistemi üzerinde bir dizin yapısı oluşturularak veri tabanından okunan ve dönüştürülen kayıtlar bu dizin yapısı altına düzenli olarak yazılmaktadır. Bunun yanında aktarımda son okunan kaydın index değerini tutmakla sorumlu bir dosya üzerinde de aktarım sonralarında güncelleme işlemi gerçekleştirilmektedir. Toplanan verilerin saklanması için Hadoop'ta yapılan çalışma yeterli görülmüştür.

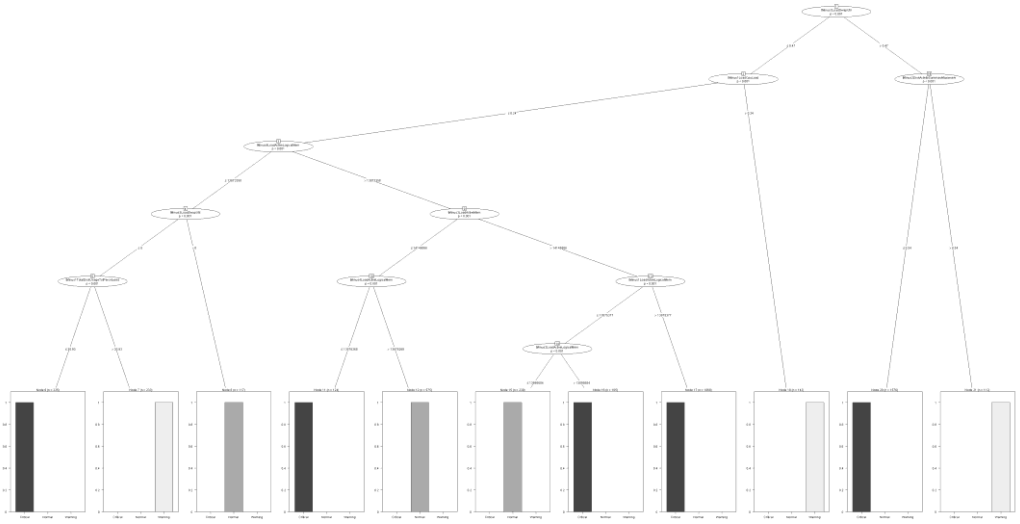
Çalışmamız kapsamında çeşitli karar ağacı modelleri, lojistik regresyon vb. sınıflandırma modelleri denenmiş ve sonuçları değerlendirilmiştir. Model girdi değişkeni olarak her bir müşteri / sistem için 10-15 adet aralığında gösterge verisi kullanılmıştır. Bu özelliklere örnek olarak toplam disk kullanım yüzdesi, işlemci kullanım yüzdesi, boş hafıza oranı, işlemci girdi-çıktı bekleme süresi, disk aktivite özeti verilebilir. Model çıktısı değişkeni ise sorun tipidir. Deneyimlerimiz ve uzman kadromuzun bilgi birikimi doğrultusunda bir sorunun çözülmesi 1 saatten daha az sürmediği için, sorunun oluştuğu andan minimum 1 saat önce gerçekleşen girdi değişkenlerinin sorun tipinin tahmin edilmesinde girdi değişkeni olarak kullanılmasına karar verilmiştir. Belirlenen girdi ve çıktılar farklı veri madenciliği teknikleri ile denenmiştir. Sonuç modeline test verisi (verinin % 30'u) için doğruluk, kesinlik ve anma performans göstergelerine göre en yüksek değere sahip olmasına bakılarak karar verilmiştir. Birer saatlik aralıklarla zamana bağlı ortalama değer değişkenleri bu modelde girdi olarak kullanılarak, çıktı değişkeni tahmin edilmektedir. Sorunun oluşmasına zamansal olarak en yakın girdi, bir saat önce tutulan girdi olduğu için, sistem bir saat sonra oluşacak sorunu tahmin etmiş olmaktadır. Çeşitli karar ağacı modelleri R Studio kullanılarak denendikten sonra bulunan en iyi karar ağacı algoritması için kod yazımı gerçekleştirilmiş, aksiyon ve tahminlerin ekran üzerine yansıtılması sağlanmıştır. İlgili kod parçası metrikler üzerinde karar ağacı kurallarını uygulamaktadır.




**Şekil 2.** EVT Sisteminin kontrol Paneli

Şekil 2'de bir ekran çıktısı görülebilen kontrol paneli üzerinde uyarı seviyeleri ve tavsiye edilen aksiyon planları izlenebilmektedir. Bu şekilde hizmet verilen firma isimleri müşteri gizliliği açısında özellikle bulanıklaştırılmıştır. Yüksek öncelikli uyarılar, gerçekleşen kritik seviyede uyarılar, olası kritik seviyede uyarılar ayrı ekranlarda takip edilebilmektedir.

Geliştirilen yazılım, Şekil 3'de bir örneği görülebilen karar ağacı modelinin sınıflandırmaları ile kritik ve uyarı seviyesindeki sorun tipleri için alınacak aksiyonları belirlemekte ve listelemektedir. Tasarlanan yapıya göre belirtilen aksiyonlar tavsiye niteliğindedir. Bu aksiyonlar sayesinde çözüme yönelik çalışmaların en hızlı şekilde başlatılmasının sağlanması hedeflenmektedir. Şekil 3'de ağaç yapısındaki düğüm miktarı ve karar ağacı model karmaşıklığına dair bir fikir elde edilebilir. Görüldüğü üzere geçmiş veriden öğrenilen modeller derinlik ve düğüm sayısı olarak çok karmaşık değildir.



**Şekil 3.** EVT Sisteme içim geliştirilen karar ağacı modellerinden bir örnek.

## 4 Sonuçlar

Yapılan çalışmalar neticesinde müşterilerde yer alan sistemlerden verilerin etkin bir şekilde transfer edilmesi sağlanmıştır. Veri inceleme ve sınıflandırma çalışmalarıyla her firma ve sunucu özelinde akan veriler ile belirlenen sorun tiplerinin (sistem çökmesi, performans düşüklüğü, CPU, hafıza problemleri vb.) uygulamaya öğretilmesi sağlanmış, veri işleme ve saklama yapısı geliştirilmiştir.

Tasarlanan önleyici sistem yapısıyla analiz edilen verilerle oluşabilecek sorunlar adreslenmiş, kontrol paneli ve raporlar yardımıyla sistem yönetimi ekibinin alınabilecek aksiyonları görüp hızlı müdahalede bulunması sağlanmıştır. Yapılan değerlendirmelerde 5694 noktalık, çoğunluğu kritik hata kayıtlarından oluşan veri kümesinde kara ağacı algoritmalarının %98 doğrulukla çalıştığı gözlemlenmiştir. Sistem yönetimi ekibi tarafından yürütülen kullanıcı kabul testleri sonrasında sistem kullanıma başlanmıştır.

Geliştirilen yapı müşterilerle de paylaşılmış ve olumlu geri bildirimler alınmıştır. Müşterilerden önce sorunlar fark edilip önleyici aksiyonlar alınarak müşteri memnuniyeti sağlanmaya başlanmıştır.

## Kaynaklar


1. Lan, Z., Zheng, Z., Li, Y.: Toward automated anomaly identification in large-scale systems. *Parallel and Distributed Systems, IEEE Transactions on*,*21*(2), pp.174-187 (2010).
2. Huang, Y., Kintala, C., Kolettis, N., & Fulton, N. D.: Software rejuvenation: Analysis, module and applications. In. *FTCS-25. Digest of Papers., Twenty-Fifth International Symposium on Fault-Tolerant Computing*, pp.381-390 (1995).
3. Alonso, J., Belanche, L., Avresky, D. R.: Predicting software anomalies using machine learning techniques. In *10th IEEE International Symposium on Network Computing and Applications (NCA)* pp. 163-170 (2011).
4. Stewart, C., Shen, K., Iyengar, A., Yin, J.: Entomomodel: Understanding and avoiding performance anomaly manifestations. *IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS)* pp. 3-13 (2010).
5. Castelli, V., Harper, R. E., Heidelberger, P., Hunter, S. W., Trivedi, K. S., Vaidyanathan, K., P.Zeggert, W. Proactive management of software aging. IBM J. Res. Dev., vol. 45, no. 2, pp. 311–332, (2001).
6. Chen, M., Zheng, A., Lloyd, J., Jordan, M., Brewer, E.: Failure Diagnosis Using Decision Trees. Proc. Int'l Conf. Autonomic Computing (ICAC), (2004).
7. Kiciman, E., Fox, A.: Detecting application-level failures in component-based Internet services. IEEE Trans. on Neural Networks, (2005).
8. Heckman, S., Williams, L.: A model building process for identifying actionable static analysis alerts. IEEE International Conference on Software Testing Verification and Validation (ICST'09), pp. 161-170, (2009).
9. Heckman, S., Williams, L.: A systematic literature review of actionable alert identification techniques for automated static code analysis. Information and Software Technology, 53(4), pp.363- 387,(2011).
10. Ekinci, Y., Duman, E.:Intelligent Classification- Based Methods in Profitability Modeling, (2015).
11. Cazenave, I.O.U, Köşlük, E., Ganiz, M.C.: An Anomaly Detection Framework for BGP. Innovations in Intelligent Systems and Applications (INISTA 2011), (2011).