Esnek Bölme ve Hiyerarşik Modeller Kullanılarak Ayrıklık Sezimi Novelty Detection using Soft Partitioning and Hierarchical Models

Tolga Ergen, Kaan Gökçesu, Mustafa Şimşek ve Süleyman S. Kozat

Elektrik ve Elektronik Mühendisliği Bölümü

İhsan Doğramacı Bilkent Üniversitesi

Ankara, Türkiye

{ergen,gokcesu,simsek,kozat}@ee.bilkent.edu.tr

Özetçe —Bu bildiride, ayrıklık sezim problemi incelenmiş ve çevrimiçi bir algoritma sunulmuştur. Algoritma ardışık olarak gözlem alır, karar verir ve parametrelerini günceller. İlk aşamada, temeldeki dağılım fonksiyonunu modellemek için bir skor fonksiyonu oluşturulmuştur. İkinci aşamada, bu skor fonksiyonu gözlemlenen verilere yönelik bir son karar vermek için kullanılmıştır. Son karar eşiklemeden sonra verilmiştir. Skor çok yönlü ve uyarlanabilir iç içe karar ağaçları ile oluşturulmuştur. İç içe esnek karar ağaçları gözlem uzayını basamaklı bir şekilde parçalara bölmek için kullanılmaktadır. Ardışık performansa bakarak ağacın her bir bileşeni uyarlanabilir bir şekilde en iyi duruma getirilmiştir. Bu zaman içindeki uyum yüksek modelleme yeteneği kazandırmaktadır, ancak, bu uyum aşırı uyma problemine neden olabilir. Bu problemi çözmek için öncelikle birçok alt ağaç üretmek için orta seviye ağaç düğümleri kullanılmış ve daha sonra bunlar birleştirilmiştir. Deneyler sunulan algoritmanın gelişkin metotlardan önemli ölçüde daha iyi olduğunu göstermiştir.

Anahtar Kelimeler—çevrimiçi, uyarlanı, ayrıklık sezimi, iç içe ağaç.

Abstract—In this paper, we study novelty detection problem and introduce an online algorithm. The algorithm sequentially receives an observation, generates a decision and then updates its parameters. In the first step, to model the underlying distribution, algorithm constructs a score function. In the second step, this score function is used to make the final decision for the observed data. After thresholding procedure is applied, the final decision is made. We obtain the score using versatile and adaptive nested decision tree. We employ nested soft decision trees to partition the observation space in an hierarchical manner. Based on the sequential performance, we optimize all the components of the tree structure in an adaptive manner. Although this in time adaptation provides powerful modeling abilities, it might suffer from overfitting. To circumvent overfitting problem, we employ the intermediate nodes of tree in order to generate subtrees and we then combine them in an adaptive manner. The experiments illustrate that the introduced algorithm significantly outperforms the state of the art methods.

Keywords—online, adaptive, novelty detection, nested tree.

I. GİRİŞ

Bu bildiride, ardışık olarak gözlemlenen veri üzerinde çalışan çevrimiçi ayrıklık sezim algoritması sunulmaktadır [1]. Her bir zaman biriminde, algoritma yeni gelen verinin aykırı olup olmadığını karar vermekte ve sonra parametreleri güncellemektedir. İlk aşamada, gözlemlenen veriye geçmiş verilere bağlı olarak ardışık bir şekilde skor atanmaktadır. Bu skora göre verinin aykırı olup olmadığına karar verilmektedir. Bu karar skoru bir eşik değeri ile karşılaştırarak yapılmaktadır [1]. Skor ataması için çok yönlü olan iç içe karar ağaçları kullanılmakta ve bu ağaçların bileşenleri ardışık performansa bağlı olarak en uygun hale getirilmektedir [2].

İki aşamalı ayrıklık sezim metotları literatürde kapsamlıca çalışılmıştır [1], [3]. Verinin dağılımını modellemek için birçok parametrelili olmayan yaklaşımlar olmasına rağmen parametrelili modeller hızlı yakınsama ve yüksek doğruluk gibi avantajlara sahiptirler [3]. Ancak, parametrik modeller farzedilen model gerçek modelden farklı olduğunda çok kötü bir performans göstermektedir [1]. Hatta, modeller uyumuna rağmen parametrik modeller hala aşırı uyum problemi yaşayabilmektedir.

Buraya kadar öncelikle yüksek derecede uyarlanabilir olan ve gözlem uzayını parçalara bölmek için kullanılan karar ağaçları sunulmaktadır. Modelleme kapasitelerini yükseltmek için her bir terminal yaprak düğüme üstel aile dağılımından bir sonsal yoğunluk fonksiyonu atanmaktadır. Her bir yaprak düğümünün bölgesinin sınırları performansa göre güncellenebilecek şekilde atanmaktadır. Bu formdaki ağaç yapısı kendini düzenleyen haritalara benzemektedir [4]. Bu bağlamda, sınırları öğrenmek Gauss yoğunluk fonksiyonun öncül ağırlıklarını öğrenmeye denk gelmektedir. Karışım modellerinin modelleme yeteneklerinin çok güçlü olduğu bilinmektedir, ancak aşırı yaprak sayısından ötürü aşırı uyum problemine sahip olabilirler [1]. Bu sorunu çözmek için yaprak düğümlere ek olarak, ağaç yapısının bütün düğümlerine bir bölge ve kendine ait bir sonsal yoğunluk fonksiyonu atanmaktadır. Bu yapı değişik derinliklerde olan birçok alt ağaç oluşturmada ve sonra bu ağaçları performansı yükseltmek için birleştirmektedir. Bu yapıda, iyi ve kötü modeller uyarlanabilir bir şekilde birleştirildiği için aşırı uyum sorunu ortadan kalkmaktadır [2].

Şekil 1: Derinliği 3 olan bir karar ağacı için katı karar sınırları.

II. AYRIKLIK SEZİM PROBLEMİ

Bu problemde, ardışık olarak $\{x_t\}_{t \geq 1}$ sinyali alınmakta ve bu sinyal $x_t \in \mathbf{R}^m$ şeklinde tanımlanmaktadır. Buradaki amaç her bir zaman indeksi t 'de, alınan sinyalin aykırı olup olmadığına karar vermektir. Kararın verilebilmesi için ardışık olarak $\{x_1, \dots, x_{t-1}\}$ verilerine bağlı olarak sonsal yoğunluk fonksiyonu $p_t(\cdot)$ oluşturulmaktadır. Daha sonra, zaman indeksi t 'de, bu fonksiyon kullanılarak x_t verisinin skoru $p_t(x_t)$ olarak belirlenir ve karar $\hat{d}_t$ olarak üretilmektedir. Son karar eşikleme kullanılarak aşağıdaki gibi elde edilmektedir:

$$p_t(x_t) \geq \tau_t, \quad (1)$$

eşitsizliği geçerli ise karar $\hat{d}_t = 0$ (normal), aksi takdirde $\hat{d}_t = 1$ (aykırı) olarak belirlenmektedir. Burada, τ_t zamana göre değişen bir eşik değeridir. Bundan sonra, karar eğer mevcut ise gerçek karar değeriyle karşılaştırılır, aksi halde denetlenmeyen bir şekilde çalışılır [3].

Ardışık dağılımı oluşturmak için karar ağaçları kullanılmaktadır. Karar ağacı dahili ve yaprak düğümlerinden oluşmuş hiyerarşik bir yapıdır. Referans [2]'deki çalışmanın aksine, burada ağacın tam olmasına gerek yoktur. x_t sinyali gözlemlendikten sonra, her bir düğüm η skorunu aşağıdaki gibi üretmektedir:

$$f_\eta(x_t) = \begin{cases} p_\eta(x_t), & \text{eğer } \eta \text{ yaprak ise,} \\ f_{\eta r}(x_t), & \sigma_\eta(x_t) \geq 0 \text{ (sağ alt düğüme git),} \\ f_{\eta l}(x_t), & \sigma_\eta(x_t) < 0 \text{ (sol alt düğüme git).} \end{cases} \quad (2)$$

Şekil 1'de gösterildiği gibi karar fonksiyonundaki $\sigma_\eta(\cdot)$ gözlemin η düğümünün hangi tarafındaki katı sınıra ait olduğunu belirlemektedir. Bu bildiride, karar sınırları için doğrusal olan ayırıcı hiper düzlemler kullanılmaktadır. Öyleki, $\sigma_\eta(\cdot)$ fonksiyonu $\sigma_\eta(x_t) = \mathbf{n}_\eta^T [x_t; 1]$ olarak verilmekte ve $\mathbf{n}_\eta$ ayırıcı hiperdüzlemin normal vektörünü temsil etmektedir. Ayrıca, yanlılık terimi dahil etmek için x_t sinyali $[x_t; 1]$ olarak genişletilmiştir. Buradaki yaklaşım genel olduğu için doğrusal olmayan ayırıcı sınırlar da kullanılabilir ancak aşırı uyum sorunundan kaçınmak için bu bildiride kullanılmamaktadır. Burada, $f_{\eta l}(x_t)$ (veya $f_{\eta r}(x_t)$) sol taraftaki (veya sağ taraftaki) alt düğümün skorunu göstermektedir. Her bir alt düğüm η , üstel aile dağılımından bir sonsal yoğunluk fonksiyonu ile aşağıdaki gibi bir dağılıma sahiptir:

$$p_\eta(x_t) = \exp(\theta_\eta^T x_t - G(\theta_\eta)) P_o(x_t).$$

Yukarıda, θ_η bir dışbükey kümeden bir değişkeni, $G(\theta_\eta)$

Şekil 2: Derinliği 2 olan bir karar ağacı için iç içe birleştirme yapısı.

yeterli istatistiği ve $P_o(x_t)$ normalleştirme değişkenini temsil etmektedir [4]. Her bir x_t için son olasılık aşağıdaki gibidir:

$$p_t(x_t) = f_1(x_t),$$

olasılığı kök düğümün skorudur. Kök düğümünden başlayarak, bu olasılık değerini bulmak için yinelenen bir şekilde, yapıklardan birine ulaşana kadar aşağı inilmektedir. Katı karar ağaçlarına ilk genişletme olarak, esnek bölme kullanılmakta ve aşağıdaki gibi hesaplanmaktadır:

$$\sigma_\eta(x_t) = 1 / [1 + \exp(\mathbf{n}_\eta^T x_t)], \quad (3)$$

denklemindeki $[x_t; 1]$ sinyali x_t olarak gösterilmektedir. Bundan sonra, her bir düğüm için

$$f_\eta(x_t) = \begin{cases} p_\eta(x_t), & \text{eğer } \eta \text{ yaprak ise,} \\ \sigma_\eta(x_t) f_{\eta l}(x_t) + (1 - \sigma_\eta(x_t)) f_{\eta r}(x_t) & \text{aksi takdirde.} \end{cases}$$

elde edilmektedir. Esnek karar ağacı için hesaplama yaprak düğümlerden başlamakta ve öyleki bütün yaprak düğümler son sonsal yoğunluk fonksiyonuna katkıda bulunmaktadır. Şekil 2'de gösterildiği gibi, son skoru elde etmek için en alttan başlayarak ağacın en üst düğümüne kadar ilerlenmektedir.

İkinci genişletme olarak bütün düğümlere üstel aile fonksiyonundan bir sonsal yoğunluk fonksiyonu atanmaktadır. Önceki durumlarda, katı veya esnek sınırlardan sadece biri ve sadece ağacın yaprakları kullanılmaktadır. Burada, dahili düğümlere dağılım atayarak daha kaba modeller de temsil edilebilmektedir. Bu dağılım atamasından sonra her bir düğüm aşağıdaki gibi tanımlanmaktadır:

$$f_\eta(x_t) = \begin{cases} p_\eta(x_t), & \text{eğer } \eta \text{ yaprak ise,} \\ \beta_\eta p_\eta(x_t) + (1 - \beta_\eta) \times \\ \quad [\sigma_\eta(x_t) f_{\eta l}(x_t) + (1 - \sigma_\eta(x_t)) f_{\eta r}(x_t)] & \text{aksi takdirde} \end{cases}, \quad (4)$$

tanımında $0 \leq \beta_\eta \leq 1$ sınırı bütün η değerleri için geçerlidir. En uygun duruma getirmek için olasılıksal bayır inişi (OBİ) [5] kullanıldığından, karışım ağırlıkları olan β_η aşağıdaki gibi yeniden tanımlanmaktadır:

$$\beta_\eta = 1 / [1 + \exp(-\alpha_\eta)], \quad (5)$$

denkleminde, $0 \leq \beta_\eta \leq 1$ sağlamak için $\alpha_\eta \in \mathbf{R}$. Son olasılık $p(x_t) = f_1(x_t)$ ile gösterilmektedir. x_t sinyali gözlemlendikten sonra son olasılık hesaplaması ağacın en altından başlamaktadır ve yapraklara ek olarak dahili düğümlerde son olasılığa katkıda bulunmaktadır.

Bu formdaki derinliği d olan bir ağaç için dahili ve terminal düğümler dahil olmak üzere $\eta = 2^{d+1} - 1$ düğüm vardır. Her bir dahili düğüm için $\{\beta_\eta, \sigma_\eta, \theta_\eta\}$ değişkenler grubu bulunmaktadır. Bunlar sırasıyla karışım katsayısı, esnek bölme parametresi ve sonsal yoğunluk fonksiyonu parametresidir. Terminal düğümler için sadece $\{\theta_\eta\}$ kullanılmaktadır.

III. ÇEVİRİMİÇİ ALGORİTMA

Bu bölümde, ardışık olarak algoritmanın parametreleri öğrenilecektir. İlk durumda, gerçek etiket bilinmemekte ve etikete (1) denkleminde göre karar verilmektedir. Eğer $\hat{d}_t = 0$ ise $\mathbf{x}_t$ gözlemi $p_t(\cdot)$ fonksiyonunu güncellemek için kullanılabilir. Eğer $\hat{d}_t = 1$ ise gözlem kullanılmaz. İkinci durumda, gerçek etiket d_t bilinmektedir. Eğer $\hat{d}_t = 0$ ise $p_t(\cdot)$ güncellenmektedir. Eğer $\hat{d}_t = 1$ ise fonksiyon güncellenmez. Gerçek etiket mevcut ise, eşik değeri de güncellenmektedir.

$p_t(\cdot)$ fonksiyonu güncellendiği zaman, performans negatif logaritma olasılık ölçütü [4] kullanılarak aşağıdaki gibi değerlendirilmektedir:

$$l_t(\mathbf{x}_t) = -\ln p_t(\mathbf{x}_t). \quad (6)$$

Parametreleri öğrenmek için OBİ [5] algoritması kullanılmaktadır. OBİ algoritmasını kullanmak için son kayıp fonksiyonun her bir parametreye göre bayırı bulunmalıdır. Şekil 2'deki esnek karar ağacı yapısı sinir ağı yapısına oldukça benzerdir [4]. Bundan dolayı, bayırları hesaplamak için geri yayılma algoritması [4] kullanılmaktadır. Geri yayılım algoritması şöyle işlemektedir: $\mathbf{x}_t$ geldiğinde, yaprak düğümlerden başlayarak $\sigma_{\eta,t}$, $\beta_{\eta,t}$, $p_{\eta,t}(\mathbf{x}_t)$ ve $f_{\eta,t}(\mathbf{x}_t)$ terimleri hesaplanmaktadır. Bu ileri yayılma aşamasıdır. Geri yayılma adımında, kök düğümden başlayarak en alt düğümlere ulaşana kadar adım adım bayır hesaplanmaktadır. Kök düğümü içeren herhangi bir dahili η için aşağıdaki denklemler geçerli olmaktadır:

$$\nabla_{\theta_\eta} l_t(\mathbf{x}_t) = \delta_{\eta,t} \beta_{\eta,t} p_\eta(\mathbf{x}_t) \left(\mathbf{x}_t - \nabla_{\theta_\eta} G(\theta_{\eta,t}) \right), \quad (7)$$

$$\begin{aligned} \partial l_t(\mathbf{x}_t) / \partial \alpha_\eta &= \delta_{\eta,t} (1 - \beta_{\eta,t}) \beta_{\eta,t} \left(p_\eta(\mathbf{x}_t) - \right. \\ &\quad \left. [\sigma_\eta(\mathbf{x}_t) f_{\eta l}(\mathbf{x}_t) + (1 - \sigma_\eta(\mathbf{x}_t)) f_{\eta r}(\mathbf{x}_t)] \right), \end{aligned} \quad (8)$$

$$\begin{aligned} \nabla_{\mathbf{n}_\eta} l_t(\mathbf{x}_t) &= \delta_{\eta,t} (1 - \beta_{\eta,t}) \times \\ &\quad (1 - \sigma_{\eta,t}(\mathbf{x}_t)) \sigma_{\eta,t}(\mathbf{x}_t) (f_{\eta l,t}(\mathbf{x}_t) - f_{\eta r,t}(\mathbf{x}_t)) \mathbf{x}_t. \end{aligned} \quad (9)$$

Yukarıda, $\delta_{\eta,t} \triangleq \partial l_t(\mathbf{x}_t) / \partial f_\eta(\mathbf{x}_t)$ tanımı geçerlidir. Geri yayılım kullanarak $\delta_{\eta,t}$ hesaplanmaktadır. Denklem (6) kullanılarak, kök düğüm için aşağıdaki eşitlik elde edilmektedir:

$$\delta_{\eta,t} = -1/p_t(\mathbf{x}_t). \quad (10)$$

Bundan sonra, üst seviyeden başlayarak alt düğümlere geri yayılma gerçekleştirilmektedir. Herhangi bir dahili düğüm için sağ ve sol alt düğümler ayırt edilmektedir. $\tilde{\eta}$ düğümünün sol alt düğümü olan η için aşağıdaki denklem yazılmaktadır:

$$\delta_{\eta,t} = \partial l_t(\mathbf{x}_t) / \partial f_{\tilde{\eta}}(\mathbf{x}_t) (1 - \beta_{\tilde{\eta},t}) \sigma_{\tilde{\eta},t}(\mathbf{x}_t). \quad (11)$$

Benzer bir şekilde, $\tilde{\eta}$ düğümünün sağ alt düğümü olan η için aşağıdaki denklem yazılmaktadır:

$$\delta_{\eta,t} = \partial l_t(\mathbf{x}_t) / \partial f_{\tilde{\eta}}(\mathbf{x}_t) (1 - \beta_{\tilde{\eta},t}) (1 - \sigma_{\tilde{\eta},t}(\mathbf{x}_t)). \quad (12)$$

Algorithm 1 Çevrimiçi Ayrıklık Sezim Algoritması

```

1:  $\alpha_{\eta,1}, \mathbf{n}_{\eta,1}, \tau_1$  için ilk değer ata
2: for  $t = 1$  to  $\dots$  do
3:   Gözlem alınması,  $\mathbf{x}_t$ 
4:   for bütün düğümler için do
5:     (3), (5), (4) denklemlerine göre  $\sigma_{\eta,t}, \beta_{\eta,t}, f_\eta(\mathbf{x}_t)$  hesapla
6:   end for
7:    $p_t(\mathbf{x}_t) = f_1(\mathbf{x}_t)$ 
8:    $\hat{d}_t = \max(0, \text{sgn}(\tau_t - p_t(\mathbf{x}_t)))$ 
9:   if ( $d_t = 0$ ) veya ( $d_t$  mevcut değilse ve  $\hat{d}_t = 0$ ) then
10:    for bütün düğümler için do
11:      (10), (11), (12) denklemlerine göre  $\delta_{\eta,t}$  hesapla
12:      (7), (8), (9) denklemlerine göre  $\nabla_{\theta_\eta} l_t(\mathbf{x}_t)$ ,
13:       $\partial l_t(\mathbf{x}_t) / \partial \alpha_\eta, \nabla_{\mathbf{n}_\eta} l_t(\mathbf{x}_t)$  hesapla
14:      (13), (14), (15) denklemlerine göre  $\theta_{\eta,t+1}$ ,
15:       $\alpha_{\eta,t+1}, \mathbf{n}_{\eta,t+1}$  parametrelerini güncelle
16:    end for
17:  end if
18:  if  $d_t$  mevcut ise then
19:    (17) denkleminde göre  $\tau_t$  değişkenini güncelle
20:  end if
21: end for

```

Yineleme terminal yaprak düğümlerde durmaktadır. Daha sonra, parametreler OBİ algoritması ile aşağıdaki gibi güncellenmektedir:

$$\theta_{\eta,t+1} = \theta_{\eta,t} - \mu_t \nabla_{\theta_\eta} l_t(\mathbf{x}_t) \quad (13)$$

$$\alpha_{\eta,t+1} = \alpha_{\eta,t} - \mu_t \partial l_t(\mathbf{x}_t) / \partial \alpha_\eta \quad (14)$$

$$\mathbf{n}_{\eta,t+1} = \mathbf{n}_{\eta,t} - \mu_t \nabla_{\mathbf{n}_\eta} l_t(\mathbf{x}_t). \quad (15)$$

Yukarıda, μ_t öğrenme hızını göstermektedir. Geri bildirim olduğu zaman, eşik değeri de OBİ kullanılarak güncellenmektedir. Kayıp fonksiyonu için $(d_t - \hat{d}_t)^2$ kullanılmakta ve böylece

$$\tau_{t+1} = \tau_t - \mu_t (d_t - \hat{d}_t) \partial \hat{d}_t / \partial \tau. \quad (16)$$

elde edilmektedir. (1) denklemindeki $\hat{d}_t$ türevlenebilir olmadığı için

$$\tilde{d}_t = 1 / [1 + \exp(-(\tau_t - p_t(\mathbf{x}_t)))] .$$

kullanılmaktadır [5]. Böylece, (16) denklemi aşağıdaki hale gelmektedir:

$$\tau_{t+1} = \tau_t - \mu_t (d_t - \tilde{d}_t) \tilde{d}_t (1 - \tilde{d}_t). \quad (17)$$

Bu adımla beraber bütün denklemler tamamlanmaktadır. Ayrıca, algoritmanın tamamı Algoritma 1'de anlatılmaktadır.

IV. SAYISAL ÖRNEKLER

Bu bölümde, İstanbul Döviz Kuru (İDK) veri kümesi kullanılarak algoritmalar karşılaştırılmaktadır. Bu veri kümesinde, 9 özelliği olan 536 zaman serisi şeklinde veri bulunmaktadır. Bu veri kümesine yapay olarak çok değişkenli Gauss sürecinden üretilmiş aykırı veri yerleştirilmiştir. Çok değişkenli Gauss sürecinin ortalaması veri kümesinin ortalaması ile aynı ve covaryansı veri kümesinin covaryansının 16 katıdır. Bu veri kümesi kullanılarak katı, esnek, iç içe karar ağaçları ve en gelişmiş algoritmalar: destek vektör veri tasviri (SVDD) [6], en yakın komşu veri tasviri (NN) [7], en büyük olabilirlik (ML) [8] merkezli ve çekirdek yoğunluk tahmini (KD) [9] merkezli ayrıklık sezim algoritmaları karşılaştırılmıştır. Önerilen algoritma olan iç içe karar ağaçları ağaçtaki dahili ve terminal düğümlerin kanılarını birleştirmektedir. Esnek karar ağaçları

Şekil 3: Ağaç derinliği ve geri bildirim olasılığı çiftleri $d = 2$, $s = 0.5$, $d = 4$, $s = 0.5$ ve $d = 4$, $s = 1$ olan katı, esnek ve iç içe karar ağaçlarının zamana bağlı eğri altı alan (EAA) performansı.

sadece sınırları ve terminal düğümleri güncellemektedir. Katı karar ağaçları ise sadece terminal düğümleri güncellemektedir. Öğrenme hızı verinin durağan olmama durumunu telafi etmek için $\mu = 1/\sqrt{t}$ olarak seçilmiştir [3]. Her bir düğümde çok değişkenli Gauss yoğunluk tahmincisi çalıştırılmıştır. İç içe ağaçlar için başlangıç kendi kendine birleştirme ağırlığı $\beta_{\eta,1} = 0.5$ ve eşik değeri $\tau_1 = 1$ olarak seçilmiştir. Her bir katman o katmana denk gelen özelliğe göre bölünmüştür ve böylece başlangıç sınırları oluşturulmuştur. Gauss yoğunluk tahmincileri için sıfır ortalama ve özdeşlik covaryansı seçilmiştir. SVDD, NN, ML ve KD'nin pencere boyutu yüzdür. Böylece pencere parametreleri adil bir şekilde öğrenmek için yeterli büyüklükte olmaktadır. Her bir kayan pencere için ML, KD ve SVDD'nin parametreleri en uygun hale getirilmiştir. ML doğal olarak kendi parametrelerini en uygun hale getirmektedir. Ayrıca, KD'nin band genişliği de en uygun hale getirilmiştir. SVDD'nin band genişliği için her bir denemede $[2^{-10}, 2^{10}]$ aralığı taranıp en uygun değer seçilmiştir.

Performans karşılaştırması için eğri altı alan (EAA) değerlendirme ölçütü kullanılmıştır. EAA ayrıklık sezicinin değişken eşik değeri tarafından örneklenmiştir. Örneklem için güncelleme kuralları değiştirilmiş ve maliyet ölçütü eklenmiştir. Parametre α yanlış pozitifin yanlış negatife oranıdır ve değişkendir. Yanlış etiketlenmiş ve aykırı veri için maliyet sırasıyla $\sqrt{\alpha}(d_t - \hat{d}_t)$ ve $(d_t - \hat{d}_t)/\sqrt{\alpha}$ olarak verilmiştir. Böylece, $\alpha > 1$ ve $\alpha < 1$ için eşik değeri sırasıyla azalmaya ve artmaya meyillidir. Bu davranış, EAA eğrisini değişik α değerleri için örneklemektedir. Şekil 3 ve 4'te, algoritmaların EAA performansları gösterilmiştir. Şekil 3'te görüldüğü gibi hem katı hem esnek karar ağaçları $d = 2$ ile daha iyi performans göstermiştir. Daha az öğrenecek parametreleri olduğu için daha hızlı yakınsamaktadırlar. İç içe karar ağaçları için dahili düğümler kullanıldığı için derinliğin bir önemi yoktur. Bu şekilde, yüksek geri bildirim daha iyi performans sağladığı ve geri bildirim bağıllılığı algoritmalar arasında değiştiği gösterilmiştir. Esnek ağaçlar $s = 0.5$ seçimi ile katı ağaçların $s = 1$ seçimi ile olan performansına yakın bir performans göstermiştir. İç içe ağaçlar ise $s = 0.5$ seçimi ile esnek ağaçların $s = 1$ seçimi ile olan performansını geçmiştir. İç

Şekil 4: Derinliği $d = 1$ olan iç içe karar ağacı, destek vektör veri tasviri (SVDD), En büyük olasılırlık (ML), çekirdek yoğunluk tahmini (KD) ve en yakın komşu veri tasviri (NN) algoritmalarının zamana bağlı eğri altı alan (EAA) performansı.

içe ağaçlar kullanmak aşırı uyum sorununu azaltmış ve diğer birleşim yapılarından daha iyi bir performans sergilemiştir. Şekil 4'te, farklı algoritmalar karşılaştırılmıştır. KD parametrik olmadığı için en yavaş yakınsamaya sahiptir. SVDD başta yavaştır ancak daha sonra ML ve NN'i hızla yakalamıştır. Her şeye rağmen, iç içe karar ağaçları bütün algoritmalarda çok daha iyi bir performans göstermiştir.

V. SONUÇLAR

Bu bildiride, yüksek derecede çok yönlü ve etkili olan iç içe ağaçlar temelli bir çevrimiçi ayrıklık sezim algoritması sunulmuştur. Ardışık performansa bakarak, ağacın karar bölgeleri ve her bir düğümdeki olasılıksal modeller dahil olmak üzere ağacın her bir bileşeni öğrenilmiştir. Aşırı uyum sorunu çok kaba modellerden çok iyi modellere kadar birçok alt ağaç üretilerek çözülmüştür.

KAYNAKLAR

- [1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection : A survey," *ACM Computing Surveys*, vol. 11, pp. 1–72, 2009.
- [2] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, 1995.
- [3] M. Raginsky, C. H. R. M. Willett, J. Silva, and R. F. Marcia, "Sequential anomaly detection in the presence of noise and limited feedback," *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5544–5562, 2012.
- [4] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. NJ: John Wiley & Sons, 2000.
- [5] A. H. Sayed, *Fundamentals of Adaptive Filtering*. NJ: John Wiley & Sons, 2003.
- [6] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.
- [7] G. G. Cabral, A. L. Oliveira, and C. B. Cahú, "Combining nearest neighbor data description and structural risk minimization for one-class classification," *Neural Computing and Applications*, vol. 18, no. 2, pp. 175–183, 2009.
- [8] H. V. Poor, *An Introduction to Signal Detection and Estimation*. NJ: Springer, 1994.
- [9] J. S. Simonoff, *Smoothing methods in statistics*. Springer Science & Business Media, 2012.