SAAS Uygulamaları için bir Anomali Algılama Sistemi

Tunç Gültekin, Özgü Can

Ege Üniversitesi Bilgisayar Mühendisliği Bölümü, 35100 Bornova, İzmir
tuncgultekin@gmail.com,
ozgucan@ege.edu.tr

Özet. SAAS ortamlarında kullanıcı şifrelerinin ele geçirilmesiyle oluşan yetkisiz oturumların tespiti ve sonlandırılması, kullanıcı güvenliği açısında oldukça önemlidir. Bunun için kullanıcıları şifreleri olmaksızın tanımlayabilecek ikincil bir yöntem gerekmektedir. İnsan fizyolojisinde sıklıkla tekrarlanan hareketler bir süre sonra kas hafızası adı verilen mekanizmaya göre istemsizce yapılmaya başlanmakta ve kişilere göre farklılık gösterebilmektedir. Bu düşünceden hareketle, çalışmada kullanıcıları web sayfaları üzerinde gerçekleştirdikleri sayfa navigasyonu, mouse hareketi ve tıklama gibi aksiyonlara göre modelleyen ve bu modellerden "anormal" oturum tespiti yapan bir uygulama çatısı sunulmaktadır.

Anahtar Kelimeler: Anomali algılama, Saas, N-Gram, Web çerçeveleri

1 Giriş

Kas hafızası diğer bir deyişle motor öğrenme, tekrarlanan hareketler sonucu elde edilen ve bir işin düşünmeksizin minimum dikkat ve maksimum verimlilik ile yapılabilmesini sağlayan hafıza biçimidir. Bisiklete binmek, enstrüman çalmak, klavye ve mouse kullanmak kas hafızası ile gerçekleştirilir [1] . İnsanların kendi alışkanlıklarına göre şekillenen kas hafızası, kimlikler hakkında da bilgi verebilir. Örneğin mouse kullanan bir kişi istemsizce ekrandaki satırları mouse işaretçisi ile izleyip tıklayabilir veya bir başkası yine istemsizce desenler çizebilir. Eğer işaretçinin izlediği yolların veya tıklanan alanların birer modeli çıkarılabilirse, kişiler de ayırt edilebilir [3].

Bu düşünceden ilham alarak bu çalışmamızda; SAAS uygulamaları için kullanıcıların gerçekleştirdiği istemli veya istemsiz aksiyonların (tıklamalar, sayfa geçişleri ve mouse hareketleri) kullanıcılara özgü modellerini çıkartan ve bu modellere göre anormal oturum tespiti yapan bir uygulama çatısı sunmaktayız. Bu çatıda, kullanıcı aksiyon dizileri, metin karşılaştırmada ve DNA eşleştirmede sıklıkla kullanılan N-Gram [2, 4] yöntemi ile temsil edilmektedir. Elde edilen N-Gram'ların, Jackard ve Cosine distance [5] yöntemleri ile geçmişte gerçekleşen aksiyon N-Gram'larına olan benzerliği hesaplanmakta ve sonuçlara göre "Normal" veya "Anormal" kullanım şeklinde kararlar üretilmektedir.

Bildirinin devamı şu bölümlerden oluşmaktadır. Bir sonraki bölümde anomali algılama üzerine yapılan önceki çalışmalar anlatılmakta ve bu çalışma ile karşılaştırılmaktadır. Kullanıcı modelleme ve anomali algılamanın detayları hakkında Yöntem başlığı altında detaylı bilgi verilmekte, uygulama çatısı ise sistem tasarımı bölümü altında anlatılmaktadır. Yapılan deneyler sonucu elde edilen bilgiler ve parametrelerin sistem performansına etkileri sırasıyla Deneysel Sonuçlar ve Sistem Parametreleri başlıklarında incelenmektedir.

2 Önceki Çalışmalar

Literatürde bir çok çalışma anomali algılama yöntemleri üzerine yoğunlaşmıştır. Bu çalışmalardan bazıları ağ paketleri üzerinden saldırı tespiti yaparken, bazıları da; web uygulamaları üzerindeki kullanıcı davranışlarından yararlanarak anormal durum algılaması yapar. Zhang ve arkadaşları yaptıkları araştırmada [11], ağ üzerindeki anormal durum tespit yöntemlerini; istatistiksel, sınıflandırıcı tabanlı, makine öğrenmesi tabanlı ve sonlu durum makinesi tabanlı olarak 4 gruba ayırmaktadır. Kullanıcı davranışları, kullanıcıların kimlikleri hakkında bilgi verdiği için web uygulamaları üzerinden yapılan anormal durum tespitinin temelini oluşturmaktadır. Velásquez'ın yaptığı çalışmada [8], web içeriği üzerindeki navigasyon dizileri, her bir sayfada geçirilen süre gibi bilgiler incelenerek kullanıcılar gruplanmakta ve davranışları analiz edilmektedir. Cadez ve arkadaşlarının çalışmasında [6] msnbc.com'dan alınan web sayfası navigasyon dizileri önceki çalışmaya benzer şekilde kullanıcı grupları ortaya çıkarmak için kullanılmıştır.

Navigasyon dizileri dışında, DOM (Document Object Model) nesnelerinin yerleşimi, bağlantıların şekli gibi web sayfalarının yapısal özellikleri de anormal durumların tespitinde kullanılabilecek özelliklerdendir. Pan'ın çalışmasında [9] web sayfalarının yapısal özelliklerinden özentilikler çıkarılmakta ve destek vektör makineleri kullanılarak, sahtecilik (phishing) amacıyla yaratılan web sayfalarının tespiti sağlanmaktadır. Birim zamanda yapılan istek sayılarının çok yüksek olduğu, aynı zamanda saldırıya da çok açık olan SAAS ortamları için anormal durum ve saldırı tespiti ayrı bir önem taşır. Nascimento ve Correia'nın araştırmasında [10], saldırı tespitinde kullanılan modeller gerçek bir SAAS uygulaması üzerinde test edilmekte ve karşılaşılan zorluklar incelenmektedir. Kullanıcı davranışlarını saldırı tespitinde kullanan ilk araştırmalardan biri olan Xie ve Yu'nun çalışmasında [7], uygulama katmanı temelli dağıtık servis reddi (application-layer-based DDoS) saldırıların erken tespiti için bir saklı yarı Markov modeli (H-sMM) tasarımı önerilmektedir.

Anlatılan çalışmalarda kullanıcı davranışları, genel saldırı tespiti ve kullanıcı profili belirleme için kullanılmakta olup bir yazılım çatısı önerilmemektedir. Biz bu çalışmamızda diğerlerinden farklı olarak; kullanıcı davranışlarından, hesap şifrelerinin ele geçirilmesi gibi yetkisiz kullanımları belirlenmesi için yararlanmakta ve bir anomali algılama çatısı önermekteyiz.

3 Yöntem

Anomali algılama sistemi; sunucu tarafında anomali algılama motoru ve istemci tarafında Javascript kütüphanesi olmak üzere iki ana bileşenden oluşmaktadır. Aktif kullanıcının, web sayfaları üzerindeki mouse hareket davranışları Javascript kütüphanesi tarafından toplanır ve bir aksiyon isteği gerçekleştiğinde, örneğin kullanıcı başka bir sayfaya geçmek istediğinde, toplanan mouse hareket bilgileri ile birlikte aksiyon tipi sunucu tarafında çalışan anomali algılama motoruna gönderilir. Burada kullanıcının aktif oturumundaki davranışları önceki oturumları ile karşılaştırılarak, normal veya anormal durum şeklinde bir karar üretilir. Aktif oturum bilgisi, kullanıcının her yeni aksiyonunda güncellenmektedir. Buna göre ilgili oturum için anomali kontrolü, sunucu tarafında belirli zaman aralıklarında bir asenkron olarak gerçekleştirilebilir. Bu şekilde periyodik asenkron kontroller ile sistem asıl uygulamanın cevap sürelerini etkilemeden anormal oturum kontrolü yapabilir. Kullanıcı sayısının fazla olduğu ortamlarda ise yatay ve dikey ölçekleme yöntemleri ile sistem performansı iyileştirilebilir. İlerleyen bölümlerde anomali algılama sisteminin bileşenleri anlatılacaktır.

3.1 Anomali Algılama Motoru

Sunucu tarafında, gönderilen kullanıcı oturum bilgilerine göre, normal veya anormal oturum şeklinde karar üreten sistemdir. Bunun için kullanıcının oturum içerisindeki aksiyonlarınının önceki oturumları ile benzerliğinden yararlanmaktadır. Aksiyon a kavramı, kullanıcın web uygulaması üzerinde gösterdiği bir davranışı (durum değişimini) ifade etmektedir ve kullanım yöntemine göre, iki web sayfası arasındaki geçişe karşılık gelebileceği gibi, aynı web sayfası üzerindeki bir "butona tıklama" gibi bir işleme de karşılık gelebilmektedir.

Kullanıcının oturum davranışlarının tanımlanmasında, her bir durumdaki mouse hareket deseni d bilgisi de aksiyon bilgisine eklenmektedir. Bir matris d ile gösterilen mouse hareket deseni, istemli mouse hareketlerini içerdiği gibi, kişiyi tanımlamada faydalı olabilecek, kas hafızası kullanılarak yapılan istemsiz hareketleri de içermektedir. Desen matrisin her bir hücresi mouse işaretçisinin, ekranın ilgili bölgesine kaç kere geldiğini belirtmektedir. Desen matrisinin oluşturulma detayları ilerleyen bölümde anlatılacaktır.

Aksiyonlar bir araya gelerek aksiyon dizilerini A yani oturumları meydana getirirler. Aktif oturum önceki oturumlar ile karşılaştırılırken, aktif oturumun aksiyon dizileri N-Gram yöntemi ile küçük gruplara ayrılmakta ardından önceki oturumların aksiyon N-Gram'ları ile karşılaştırılmaktadır. Durum geçiş bilgisi aynı olan iki farklı aksiyon veya aksiyon grubu, mouse hareket desenine bakılmaksızın "Eş" olarak kabul edilmiştir. Eşleşen aksiyon N-Gram'ları sayılarak, Jackard Distance yöntemi ile oturumlar arasında benzerlikler hesaplanmakta ve bu benzerliklerin ortalaması alınmaktadır. Elde edilen ortalama, karşılaştırılan oturumun önceki oturumlar ile arasındaki aksiyon davranışlarının benzerliğini vermektedir.

Örnek aksiyon dizileri:

$A_1 = \{a_{a-b}, a_{b-c}, a_{c-g}, a_{g-k}, a_{k-t}, a_{t-b}\}$
$A_2 = \{a_{a-b}, a_{b-c}, a_{c-a}, a_{a-k}, a_{k-t}, a_{t-b}\}$

Aksiyon dizileri için 2'li N-Gram'lar:

$A_1 = \{\{a_{a-b}, a_{b-c}\}, \{a_{b-c}, a_{c-g}\}, \{a_{c-g}, a_{g-k}\}, \{a_{g-k}, a_{k-t}\}, \{a_{k-t}, a_{t-b}\}\}$
$A_2 = \{\{a_{a-b}, a_{b-c}\}, \{a_{b-c}, a_{c-a}\}, \{a_{c-a}, a_{a-k}\}, \{a_{a-k}, a_{k-t}\}, \{a_{k-t}, a_{t-b}\}\}$

Eşleşen 2'li N-Gram'lar:

$\{a_{a-b}, a_{b-c}\}, \{a_{k-t}, a_{t-b}\}$

Jackard Distance: (2 / 5)

mouse hareket desenlerinden alınan bilgilere göre sonuçları iyileştirmek için, eşleşen aksiyonların desen matrisleri vektörel hale getirilip, Cosine distance yöntemi ile benzerlikleri hesaplanmaktadır. Ardından yine bu benzerliklerin ortalaması alınmaktadır. Elde edilen ortalama, karşılaştırılan oturumun önceki oturumlar ile eş kabul edilen aksiyonlarının bireysel benzerliğini vermektedir.

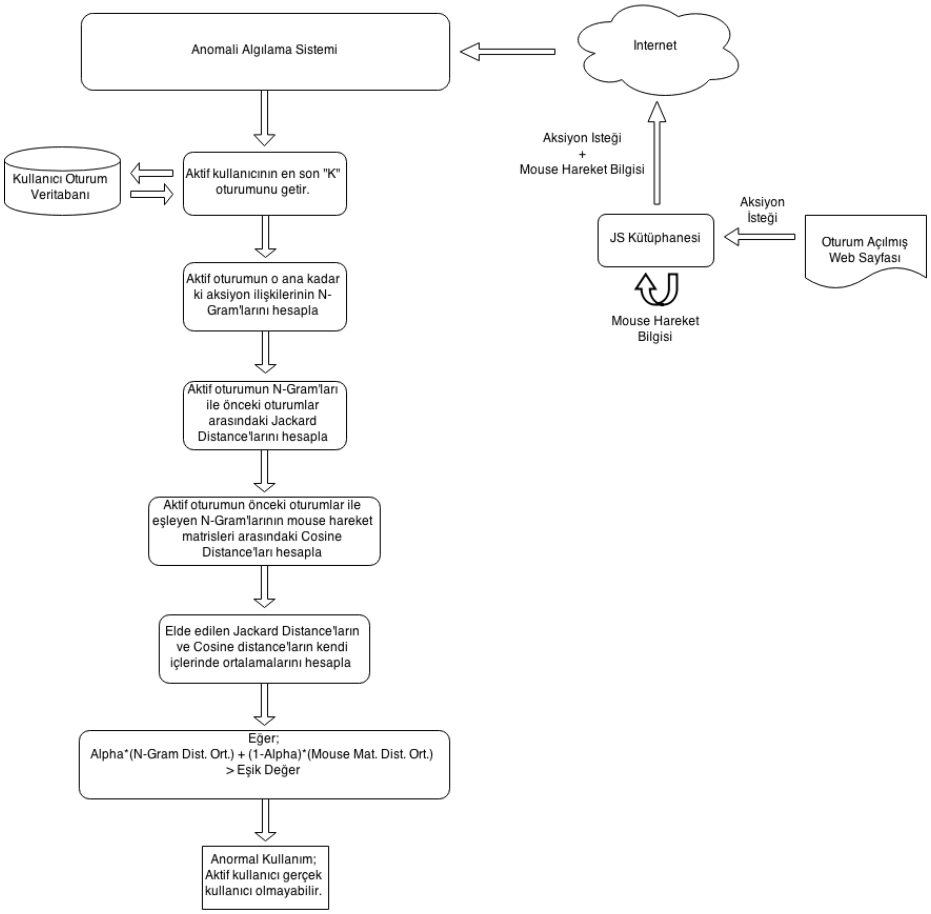
Oturum benzerlik skoru:

$$\frac{1}{\alpha * \beta + (1 - \alpha) * \gamma}$$

α = Aksiyon Dizi Benzerliği / Mouse Hareket Benzeriği Ağırlığı
β = N-Gram Jackard Distance Ortalaması
γ = Mouse Desen Matrisleri için Cosine Distance Ortalaması

Benzerlik skoru belirli bir eşik değerin altında olan oturumlar, anormal olarak kabul edilir. Skor üretiminde, aksiyon geçişlerine mi, yoksa aksiyonlarının bireysel benzerliğine mi daha çok ağırlık verileceği Alpha parametresi ile belirlenmektedir. Bunun sayesinde sistem uygulamaya özgü şekilde ayarlanabilmektedir. Örneğin; aksiyon sayısının az olup, tekil sayfa aktivitesinin daha önemli olduğu uygulamalarda Alpha değeri 0'a yakın tutulmalıdır. Oturum karşılaştırmaları yapılırken eskiyen kullanıcı alışkanlıklarından etkilenmemek için, kullanıcının yalnızca en yeni K adet oturumu kullanılmaktadır. Karar üretim süreci Şekil - 1 'de anlatılmaktadır.

Şekil - 1: Anomali algılama sistemi.



3.2 Javascript Kütüphanesi

İstemci tarafında çalışarak, kullanıcının aksiyon isteklerini, anomali algılama motoruna bildiren ve her aksiyon için t birim sürede bir mouse hareket koordinatlarını kaydeden sistemdir. Bu koordinatlardan sunucu tarafında d matrisi ile gösterilen mouse hareket deseni hesaplanır. r ve c matrisin satır ve sütun sayıları olacak şekilde, tarayıcı ekranı yatayda r, dikeyde de (c hücreye bölünmüş kabul edilerek, kaydedilmiş olan mouse işaretçi konumlarına karşılık gelen matris hücrelerinin değeri bir arttırılır. Sonuçta matrisin her bir hücresi, mouse işaretçisinin ekranın ilgili bölgesine kaç kere geldiğini göstermektedir. Bu işlemin detayları, Algoritma - 1'de yer almaktadır. Kullanıcı tarafından bir aksiyon isteği olduğunda matrisin son hali aksiyon bilgisi ile birlikte anomali algılama motoruna gönderilir. Matrislerin karşılaştırılması sırasında uç değerlerin eğilim

yaratmaması için minimum ve maksimum değerlere göre normalizasyon yapılmaktadır. Örnek bir ekran ve mouse desen matrisi Şekil 2.a ve 2.b'de görülebilir.

Algoritma 1 JsActionCapturer

```
1: xSeperation = getScreenXSize()/c
2: ySeperation = getScreenYSize()/r
3: d ⇐ c by r zero matrix
4: for each time t do
5:    x = getMouseXCoord()
6:    y = getMouseYCoord()
7:    for i = 0 to c − 1 do
8:       if i==0 then
9:          xPos = 0
10:      else
11:         xPos = x/(i ∗ xSeperation)
12:      end if
13:      for j = 0 to r − 1 do
14:         if j==0 then
15:            yPos = 0
16:         else
17:            yPos = y/(j ∗ ySeperation)
18:         end if
19:         d[xPos][yPos] = d[xPos][yPos] + 1
20:      end for
21:   end for
22: end for
```

4 Sistem Tasarımı

Sunucu tarafında çalışan anomali algılama motoru, .Net C# kullanılarak geliştirilmiştir ve aşağıdaki sınıflardan oluşmaktadır.

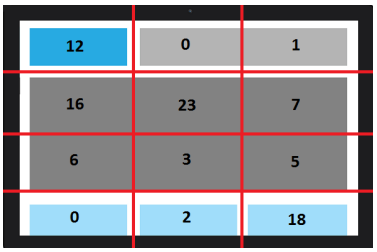
- *User*
- *MouseMoveTrace*
- *Action*
- *ActionSequence*
- *AnomalyDetectorSession*
- *AnomalyDetector*
- *AnomalyDetectorHttpHandler*
- *Utils*

Bu sınıflara ait sınıf diyagramları Şekil - 3 'de görülebilir.

User: Sistemde her kullanıcıya ait tanımlayıcı bilgileri tutan sınıftır. Tanımlayıcı olarak "UserName" (kullanıcı adı) kullanılmaktadır. User nesneleri, UserName değerlerine bakılarak karşılaştırılmaktadır bunun için; .Net ile birlikte ge-

Şekil - 2: Örnek ekran ve mouse desen matrisi.

(a) Örnek ekran



(b) mouse desen matrisi

len ön tanımlı "IEquatable" arayüzü uygulanmış ve "Equals" yordamı içerisinde UserName karşılaştırması yapılmıştır.

MouseMoveTrace: Javascript Kütüphanesi tarafından gönderilen mouse hareket koordinatlarının tutulduğu sınıftır.

Action: Kullanıcının web uygulaması üzerinde gerçekleştirdiği bir operasyonu bir durum değişimi olarak belirtir ve ilk durumdaki mouse hareket desenini üzerinde barındırır. Nesne karşılaştırmalarında kullanmak üzere, .Net ile birlikte gelen ön tanımlı "IEquatable" arayüzü uygulanmış ve "Equals" yordamı içerisinde "State1" ve "State2" 'ye göre karşılaştırma yapılmıştır.

ActionSequence: Kullanıcının bir oturum içerisinde gerçekleştirdiği aksiyon dizilerini tutar.

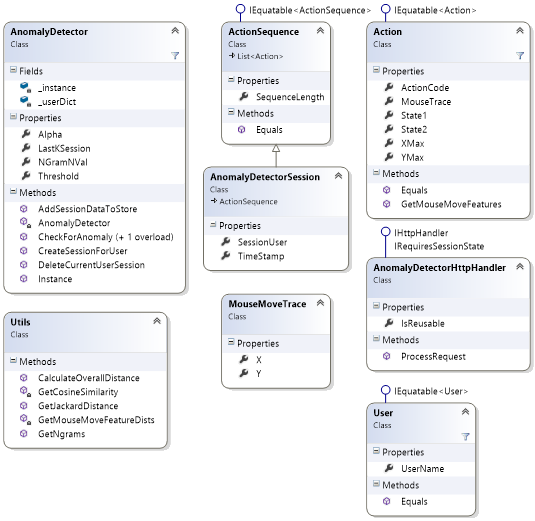
AnomalyDetectorSession: "ActionSequence" sınıfından türetilen sınıf, kullanıcının bir oturumunu temsil eder, aksiyon dizileri ile birlikte kullanıcıyı tanımlayan bilgileri de içerisinde barındırır.

AnomalyDetectorHttpHandler: Javascript kütüphanesi tarafından XhrPostları ile Json formatında gönderilen aksiyon isteklerini işleyerek aktif oturumun aksiyon dizilerine ekleyen sınıftır. .Net'in IHttpHandler arayüzünden türetilmiştir.

Utils: Jackard ve Cosine Distance'ların hesaplanması, mouse koordinatlarından desen matrislerin yaratılması ve aksiyon N-Gram'larının oluşturulması gibi ortak olarak kullanılan temel işlevleri sağlayan sınıftır.

AnomalyDetector: Sistemin temel bileşenidir, "Singleton" tasarım deseni kullanılarak geliştirilen sınıf; sistem parametreleri ile birlikte çeşitli "public" yor-

Şekil - 3: Anomali algılama motoru sınıf diagramı.



damları barındırır;

Javascript kütüphanesi AnomalyDetectorJsLib tipinde bir sınıf olarak tasarlanmıştır. Web uygulamasında anomali analizine dahil edilmek istenen her Aksiyon (bağlantıya tıklama veya bir Javascript fonksiyonu çağırma), AnomalyDetectorJsLib'in bir nesnesinin "DoAction" fonksiyonu üzerinden çağırılmalıdır. Bu fonksiyon bir sonraki aksiyonun "Id" 'si ile birlikte aksiyon içeriğini parametrik olarak alır, aksiyon bilgisini Json formatında XhrPost olarak sunucu tarafına gönderir ve aksiyonun kendisini gerçekleştirir. AnomalyDetectorJsLib nesneleri aynı zamanda, belirtilen zaman aralığında bir kullanıcının mouse hareket koordinatlarını bir liste olarak kaydeder. Bu koordinatlar her aksiyon isteğinde aksiyon bilgisi ile birlikte sunucuya gönderilir.

5 Deneysel Sonuçlar

Anomali algılama sisteminin verimliliği, uygun bir gerçek veri seti bulunamadığından, yapay kullamıcılar ve kullanıcı davranışlarını içeren bir veri seti yaratılarak analiz edilmiştir. Bunun için rastgele tekdüze dağılış kullanılarak;

- 20 farklı aksiyon ismi tanımlanmıştır.
- Aksiyonlar arası geçiş olasılıklarını içeren 20x20'lik 10 farklı matris tanımlanmıştır. Buradaki her matris farklı birer kullanıcı profiline karşılık gelmektedir.
- Her kullanıcı için minimum 15 maksimum 20 adet oturum yaratılmıştır.

- Her bir oturumda ilgili kullanıcının geçiş olasılıkları matrisi kullanılarak yaratılan minimum 8, maksimum 10 aksiyon yer almaktadır.

Yaratılan veri seti kullanıcı oturum listesine eklendikten sonra, test için aynı geçiş olasılıkları matrisleri kullanılarak yeni oturumlar üretilmiş ve bunların normal oturum olarak gösterilip gösterilmediğine bakılmıştır. Benzer şekilde farklı kullanıcıların test oturumları birbirleri ile değiştirilerek anormal oturumların algılanıp algılanmadığına da bakılmıştır. Güvenilir test sonuçları elde etmek için 100 farklı veri seti türetilip bunların doğruluk sonuçlarının ortalaması alınmıştır. Buna göre sistemin doğruluğu %83.21 olarak görülmektedir.

6 Sistem Parametreleri

Sistem 3 adet dış parametreye sahiptir, bunlar; "Anomali Eşik Değeri T" , Aksiyon geçişi / Aksiyon benzerliği oranını belirleyen "Alpha" ve en son kaç oturum ile karşılaştırma yapılacağını belirleyen "K" değerleridir. 0 ile 1 aralığında olan Anomali Eşik Değeri için düşük değerler verildiğinde sistem tüm oturumları anormal olarak algılamaktadır, yüksek değerlerde ise anormal oturum bulma oranı düşmektedir. Alpha parametresi sistemin çalıştırılacağı web uygulamasına göre belirlenmelidir, Alpha 1'e yakın iken sistem kullanıcı Aksiyonlarının sıralama benzerliğine daha çok önem vermektedir, sıfıra yaklaştığında ise aksiyonlardaki mouse hareket desenlerinin benzerliği önem kazanır.

İç parametre olarak kabul edilen N-Gram'lardaki N sayısı 3 olarak alınmıştır, bu sayının üstünde elde edilebilecek N-Gram kombinasyonları (aksiyon sayısı)^N şeklinde arttığı için karşılaştırılması gereken N-Gram sayıları da artmaktadır. Daha düşük değerlerde ise sistemin doğru algılama oranı düşmektedir. Farklı parametre birleşimleri, 100 farklı set üzerinde test edilerek, yapay veri setleri için en iyi parametre yapılandırması şu şekilde belirlenmiştir; Anomali Eşik Değeri T = 0,88 Alpha = 0,9 K = 10

Buna göre yapay veri setleri üzerinde kullanıcıların mouse hareket davranışlarının iyi bir şekilde modellenemediği ve sistemin yüksek Alpha değerlerinde başka bir deyişle aksiyonların dizi benzerliğine önem veren yapılandırmada iyi çalıştığı görülmektedir. Gerçek bir sistemde en uygun parametre değerlerini belirlemek için; bir süre oturum örneklemesi yapılarak farklı parametre kombinasyonlarının başarı oranları hesaplanabilir ve bunlar üzerinde optimizasyon yapılabilir.

Parametrelerin farklı değerlerine göre elde edilen doğru algılama oranları Şekil - 4 'de görülmektedir. Şekil - 4.a 'ya göre Alpha değerindeki artışa bağlı olarak doğru algılama oranı artmaktadır. Alpha değerinin artması navigasyon benzeriğinin katkısını arttırdığı için, test veri setinde navigasyon benzerliklerinin daha belirleyici olduğu söylenebilir. Şekil - 4.b 'de karşılaştırılacak oturum sayısını belirleyen K değerinin artması doğrululuk oranını arttırsa da etkisinin düşük olduğu görülmektedir. Şekil - 4.c 'de ise eşik değerindeki artışın sonuçları iyileştirdiği söylenebilir fakat 0.88 değerinden sonra sonuçlar sabit kalmaktadır. Bu testler için, analiz edilen parametre dışındaki parametrelerde belirtilmiş olan optimum değerleri kullanılmıştır. Anomali Eşik Değeri T düşük iken, sistem tüm

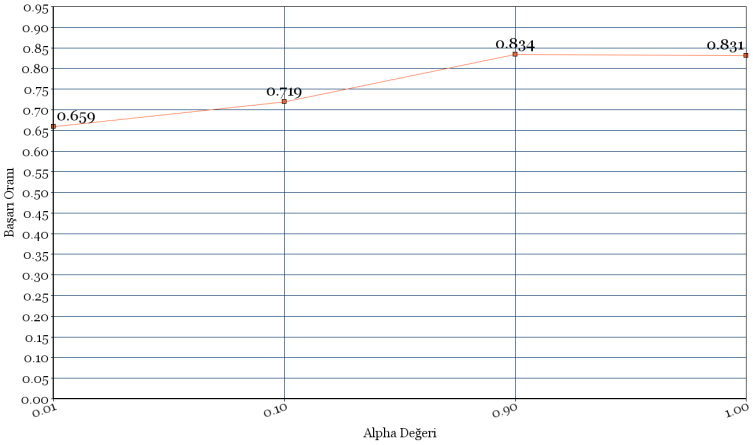
oturumları anormal olarak kabul etmekte ve başarı oranı düşmektedir, benzer şekilde yüksek değerde ise tüm oturumlar normal olarak etiketlenmektedir. Alpha'nın düşük değerlerinde sistem başarısının oldukça düştüğü gözlenmektedir, bunun sebebi; yapay veri setinin mouse hareket desenleri açısından daha düşük ayırt edilebilirliğe sahip olmasıdır. Karşılaştırmalarda kullanılacak olan oturum sayısını belirleyen K parametresinin değişimi sistem başarısını kayda değer şekilde etkilememiş gibi görünse de, bu parametre gerçek veride zaman içinde değişebilecek kullanıcı davranışlarını yakalamak açısından faydalı olacaktır. Örneğin bir SAAS kullanıcısı zaman içerisinde farklı sayfa navigasyon davranışları veya farklı mouse hareket alışkanlıkları geliştirebilir, buna göre sistemin kullanıcının son oturumlarını göz önüne alarak anomali skorları üretmesi, doğruluğu arttıracaktır.

7 Sonuç

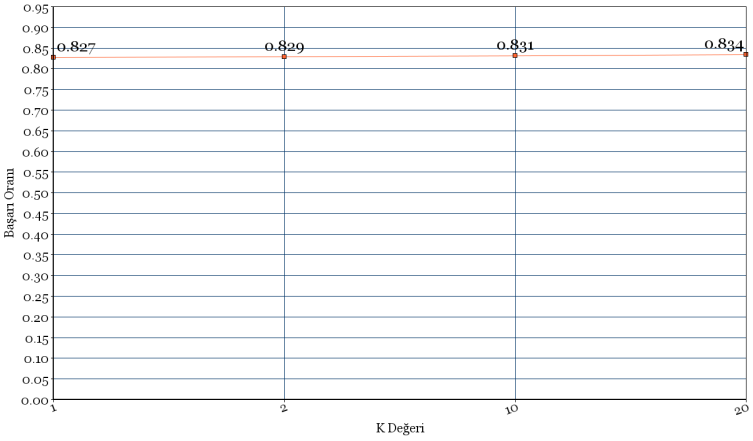
Bu çalışmada kullanıcıların SAAS uygulamaları üzerinde yaptıkları sayfa navigasyonu, tıklama ve mouse hareketi gibi aksiyonlara göre kendilerine özgü kullanım modellerini çıkaran ve bu modelleri, "Anormal" oturum tespitinde kullanan bir uygulama çatısı sunulmuştur. Sayfalar üzerinde ve sayfalar arasında gerçekleştirilen tüm aksiyonlar N-Gram'lar ile modellenmekte ve önceki aksiyonlar ile karşılaştırılmaktadır. Bunun yanında mouse hareketleri de matrisler ile temsil edilmekte ve aksiyon bilgileri eşleşen N-Gram karşılaştırma sonuçları, matris benzerlikleri ile daha da iyileştirilmektedir. Çalışma özellikle SAAS ortamlarında kullanıcı şifrelerinin 3. Şahıslar tarafından ele geçirilmesi durumunda, sistem kullanım alışkanlıklarına göre "Anormal" oturumların algılanması ve sonlandırılması için önemlidir. Sistemi test etmek için rasgele tekdüze dağılış ile yapay kullanıcı oturumları veri-seti üretilmiş ve sistemin %83'ün üzerinde başarı üretmekte olduğu gözlenmiştir.

Şekil - 4: Parametre Analizi

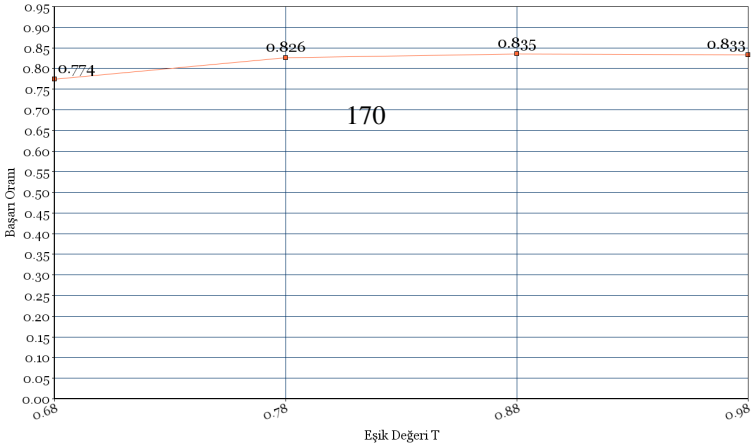
(a) Alpha / Sistem Başarı Oranı



(b) K / Sistem Başarı Oranı



(c) Eşik Değeri T / Sistem Başarı Oranı



Kaynaklar

1. Krakauer, J.W., Shadmehr, R.: Consolidation of Motor Memory. Trends in Neurosciences. 29, 58–64 (2006)
2. William B. Cavnar and John M. Trenkle: N-Gram-Based Text Categorization. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175. (1994)
3. William B. Cavnar and John M. Trenkle: Prediction of Human Behaviors in the Future Through Symbolic Inference. International Conference on Robotics and Automation (ICRA), pp. 1970–1970. IEEE, Shanghai (2011)
4. Andrija TomoviÄĞ, Predrag JaniÄIJiÄĞ, Vlado KeÅąelj: N-Gram-based Classification and Unsupervised Hierarchical Clustering of Genome Sequences. Computer Methods and Programs in Biomedicine, vol. 81, pp. 137–153. (2006)
5. Singhal, Amit: Modern Information Retrieval: A Brief Overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering , vol. 24, pp. 35–43. (2001)
6. I. Cadez, D. Heckerman, C. Meek, P. Smyth and S. White: Visualization of Navigation Patterns on a Web Site Using Model-based Clustering. Knowledge Discovery and Data Mining , pp. 280. (2000)
7. Y. Xie and Shun-Zheng Yu: A Large-Scale Hidden Semi-Markov Model for Anomaly Detection on User Browsing Behaviors. IEEE/ACM Transactions on Networking , vol. 17, pp. 54–65. (2009)
8. J. VelÃąsquez, H. Yasuda, and T. Aoki: Combining the Web Content and Usage Mining to Understand the Visitor Behavior in a Web Site. In Proceedings of 3rd IEEE International Conference on Data Mining, pp. 669–672. (2003)
9. Y. Pan, X. Ding: Anomaly Based Web Phishing Page Detection. In Proceedings of 22nd Computer Security Applications Conference, pp. 381–392. (2006)
10. G. Nascimento, M. Correia: Anomaly-based Intrusion Detection in Software as a Service. In Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, pp. 19–24. (2011)
11. W. Zhang, Q. Yang, Y. Geng: A Survey of Anomaly Detection Methods in Networks. In Proceedings of the International Symposium on Computer Network and Multimedia Technology, pp. 1–3. (2009)