EŞLER ARASI AĞLARDA GÜVEN YÖNETİMİNİN GENETİK PROGRAMLAMA İLE SAĞLANMASI

TRUST MANAGEMENT IN PEER-TO-PEER NETWORKS USING GENETIC PROGRAMMING

UĞUR ERAY TAHTA

YRD. DOÇ. DR. AHMET BURAK CAN
Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin
Bilgisayar Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2014

UĞUR ERAY TAHTA'nın hazırladığı **"Eşler Arası Ağlarda Güven Yönetiminin Genetik Programlama İle Sağlanması"** adlı bu çalışma aşağıdaki jüri tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Ebru AKÇAPINAR SEZER
Başkan :. . . . . .

Yrd. Doç. Dr. Ahmet Burak CAN
Danışman :. . . . . .

Yrd. Doç. Dr. Erhan MENGÜŞOĞLU
Üye :. . . . . .

Yrd. Doç. Dr. Murat AYDOS
Üye :. . . . . .

Öğr. Gör. Dr. Fuat AKAL
Üye :. . . . . .

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ
Fen Bilimleri Enstitüsü Müdürü

Hayat..

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

. . . . . /. . . . . /. . . . . .

Uğur Eray TAHTA

ÖZET

EŞLER ARASI AĞLARDA GÜVEN YÖNETİMİNİN GENETİK PROGRAMLAMA İLE SAĞLANMASI

Uğur Eray TAHTA
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Danışmanı: Yrd. Doç. Dr. Ahmet Burak CAN
İkinci Tez Danışmanı: Yrd. Doç. Dr. Sevil ŞEN
Ocak 2014, 81 Sayfa

Eşler arası sistemler, her kullanıcıya kolay paylaşım ve açık erişim olanağı sağlaması sebebiyle yaygın olarak kullanılmaktadır. Fazla sayıda kullanıcıya hitap eden eşler arası sistemlerde, sistemi kötü amaçlı kullanan kullanıcılar da bulunabilmektedir. Bu durum ise eşler arası sistemlerde güven yönetimini sağlamayı gerektirmektedir. Sistemde var olan kötü niyetli kullanıcıların sistemden uzaklaştırılması adına birçok yöntem uygulanmıştır. Yöntemlerin temel amacı kötü niyetli kullanıcıları tespit etmek ve onlar ile etkileşime girilmesini önlemektir.

Yapılan tez çalışması kapsamında eşler arası sistemlerdeki güven yönetimi, saldırganlara karşı eğitilebilen ve evrimleşerek daha iyi çözümler sunabilen bir model ile sağlanmıştır. Genetik programlama yardımı ile evrimleşen ve kötü niyetli kullanıcıların karakteristiklerini öğrenerek

sistemden uzaklaştıran bir model oluşturulmuştur. Kullanıcıların doğrudan birbirleri ile olan etkileşimleri ve komşularından aldıkları tavsiyeler üzerine kurulu olan model sayesinde sistemdeki kötü niyetli kullanıcıların yaptığı saldırılar engellenmeye çalışılmıştır. Farklı durumlara ve saldırı türlerine göre eğitilen model çeşitli ortamlarda test edilmiş ve başarılı sonuçlara ulaşılmıştır.

ANAHTAR SÖZCÜKLER: Eşler Arası Ağlar, Güven Yönetimi, Genetik Programlama, Evrimsel Hesaplama

ABSTRACT

TRUST MANAGEMENT IN PEER-TO-PEER NETWORKS USING GENETIC PROGRAMMING

Uğur Eray TAHTA
Master of Science, Department of Computer Engineering
Supervisor: Asst. Prof. Dr. Ahmet Burak CAN
Co-Supervisor: Asst. Prof. Dr. Sevil ŞEN
January 2014, 81 Pages

Peer-to-peer systems are used commonly by virtue of enabling easy resource sharing and open access to every user. Peer-to-peer systems with large number of peers may also contain peers that use the systems maliciously. This situation makes the trust management necessary in the peer-to-peer systems. Many methods have been applied to remove existing malicious peers from the system. Main purpose of these methods is to identify the malicious peers and prevent interaction with them.

Within the context of this thesis, trust management in the peer-to-peer systems is provided with a model which trains and improves itself according to the attackers. With the help of genetic programming, a model which evolves and removes malicious peers by detecting their characteristics is developed. Using the model based on peers' direct interactions with

each other and recommendations from neighbors, attacks of malicious peers in the system are tried to be prevented. The model trained for different situations and attack types, is tested in various configurations and successful results are obtained.

KEYWORDS: P2P Networks, Trust Management, Genetic Programming, Evolutionary Computation

TEŞEKKÜR

Tez konusunu belirlemenmesinde ve tez süresince tez ile ilgili bildiri ve tez metni konusundaki düzenlemelerinde büyük yardımı olan hocalarım Sayın Yrd. Doç. Dr. Ahmet Burak CAN'a ve Sayın Yrd. Doç. Dr. Sevil ŞEN'e,

Tez metnini inceleyerek biçim ve içerik bakımından son halini almasına yardımcı olan Sayın Doç. Dr. Ebru AKÇAPINAR SEZER'e, Sayın Yrd. Doç. Dr. Erhan MENGÜŞOĞLU'na, Sayın Yrd. Doç. Dr. Murat AYDOS'a ve Sayın Öğr. Gör. Dr. Fuat AKAL'a,

Lisans ve yüksek lisans boyunca çok değerli bilgiler öğrendiğim üzerimde emeği geçen tüm hocalarıma,

Manevi desteğini ve fikirlerini esirgemeyerek daima destek olan değerli dostlarıma ve arkadaşlarıma,

Her koşulda beni destekleyen ve daima yanımda olan sevgili aileme,

teşekkür ederim.

İÇİNDEKİLER

ŞEKİLLER

ÇİZELGELER

Sayfa

SİMGELER VE KISALTMALAR

P2P Peer-to-peer
DHT Distributed hash table
GP Genetik programlama
GA Genetik algoritma

ALGORİTMALAR

1 GİRİŞ

Günümüzde, eşler arası sistemlerdeki gelişmeler ve bu sistemlerin kullanım oranlarındaki artış sebebiyle güven yönetimi gittikçe önem kazanmıştır. Kötü niyetli kullanıcıların olması ve bunların fark edilebilmesinin zorlaşması, bu sistemlerde başlı başına bir problem teşkil etmeye başlamıştır. Bu problemin çözümü adına da farklı modeller içeren güven yönetim sistemleri ortaya atılmıştır.

Güven yönetimi sistemlerinin temeli kötü niyetli olan kullanıcıların sistemden uzaklaştırılması ve sistem kaynaklarından yararlanmasının engellenmesi amacına dayanmaktadır. Fakat bunu başarmak, sistemin sınırlarının bilinmemesi sebebiyle gerçekten güç bir probleme dönüşmektedir. Çünkü kötü niyetli bir kullanıcı ile masum kullanıcıyı kesin çizgilerle birbirinden ayıran bir sınıflandırma yaklaşımı bulmak güçtür. Bu sebepten dolayı yaklaşık bir sınıflandırma yapan yaklaşımlar daha çok kullanılmaktadır.

Merkezi otoriteler yardımıyla güven yönetimini sağlamak, kullanılan en temel yöntemlerden biridir. Buna en bilindik örnek olarak ”eBay” verilebilir. Kullanıcılar her işlemlerinden sonra birbirlerini oylamakta ve diğer kullanıcılar hakkındaki görüşlerini bildirmektedirler. Bu oylar ve görüşler merkezi bir sistemde toplanmakta ve bir kullanıcı işlem yapmadan önce kendisine gösterilmektedir. Bu sayede merkezi otoriteye güvenen kullanıcı, gördüğü bilgiler ışığında işlemlerini gerçekleştirmekte ve sistemin güven yönetimi sağlanmaktadır. Fakat eşler arası sistemler, doğaları ve varoluş amaçları nedeniyle genel olarak merkezi bir otorite içermezler. Bu tip sistemlerde, kullanıcılar ya kendi güvenlerini kendileri saklar ve yönetirler ya da sistem üzerinde saklanan güven bilgilerini sorgulayarak öğrenirler [3, 4]. Gnutella temel ve saf bir ağ olarak tanımlanabilir [5]. Kullanıcılar genel olarak aradıkları bir dosyayı oluşturdukları bir sorgu ile tüm ağa gönderirler

ve gelen cevaplar sayesinde dosya değişimini tamamlarlar. Fakat bu yapı bedavacı kullanıcılara (*free riders*) uygun bir ortam sağlamaktadır [6].

Güveni sağlamak adına tavsiyelere dayalı itibar değeri kullanımı da eşler arası sistemlerde kullanılan bir yöntemdir. Bu tip sistemlerde, tüm kullanıcılar diğer kullanıcılar hakkındaki geçmiş deneyimleri sayesinde oluşturdukları bilgiyi tutarlar [4, 7, 8]. Bir kullanıcı hakkında fikir sahibi olabilmek için ağa sorgu gönderirler ve o kullanıcı hakkında diğer kullanıcıların tavsiyelerini alırlar. Aldıkları tavsiyeler doğrultusunda kullanıcının kötü niyetli olup olmadığına karar verirler. Bazı modeller güveni yönetmek amacı ile Dağıtık Özet Çizelgesi (*Distributed Hash Table - DHT*) kullanmaktadır. Her bir kullanıcı diğer kullanıcılar ile olan etkileşimlerini ve sağladıkları bilgileri bu DHT'leri kullanarak saklamaktadırlar [3, 9, 10]. Bu sayede her bir kullanıcı hakkında global güven bilgisine erişilebilmektedir.

Eşler arası sistemlerde, belirsiz varsayımlar ve sisteme katılımın her tür kullanıcıya açık olması nedeniyle güven yönetimini sağlamak bir hayli zor olmaktadır. Bu problemi aşmak adına birçok olası çözüm bulunmakta fakat bu çözümleri sabit formüller kullanarak yapmak problemin doğası ve özelliklerine aykırı durum oluşturmaktadır. Zor ve geniş çözüm kümesine sahip problemlerde etkili çözümler üretebilen makine öğrenmesi teknikleri eşler arası sistemlerde de kullanılabilmektedir. Eşler arası sistemlerin geniş ölçekli ve nitelendirilebilir olmayan yapısı dolayısıyla, güven yönetimini sağlayacak olan model, kötü niyetli davranışları sistemden uzaklaştırmak için var olan bilgiler ışığında makine öğrenmesi tekniklerini kullanarak eğitilebilir. Bu sayede sabit bir formül ile çözüm üretmek yerine, problemi oluşturan unsurların davranış şekillerine göre oluşturulmuş esnek çözümler oluşturulabilmektedir.

Bu tez kapsamında, eşler arası sistemlerde güven yönetimini sağlamak adına genetik programlama tekniği kullanılarak genetik güven yönetim modeli oluşturulmuş ve gerçekleştirilmiştir. Oluşturulan modelde kullanıcılar

iki türe ayrılmaktadırlar. Bunlar kötü niyetli kullanıcılar ve masum kullanıcılardır. Model genel olarak, genetik programlama yardımıyla tez çalışması kapsamında oluşturulmuş özniteliklere (*features*) göre toplanan bilgilerden faydalanarak iyi ve kötü niyetli kullanıcıların karakteristiklerini belirlemektedir. Her bir kullanıcı iki tür özniteliği toplamaktadır; etkileşimler (*interactions*) ve tavsiyeler (*recommendations*). Kullanıcılar diğer kullanıcılar ile yaptıkları geçmiş etkileşimlerini saklamakta ve diğer kullanıcılar hakkında tavsiyeler toplamaktadırlar. Tüm bu sakladıkları ve topladıkları bilgiler sayesinde bir kullanıcının türünü tahmin etmeye çalışırlar. Bilgiler genetik programlamaya öznitelik olarak verilir ve genetik bir çözüm bulunmaya çalışılır. Genetik programlamanın yardımı ile güven değerini hesaplayacak ve kullanıcının türünü belirlemeye yardımcı olacak denklemler oluşturulmuştur. Bu sayede farklı problemler ve saldırılar üzerinde esnek çözümler bulunmuştur. Bir kullanıcı başka bir kullanıcı ile iletişime geçmeden önce bulunan genetik denklemi kullanarak güven değerini hesaplamaktadır. Hesaplanan güven değerleri sayesinde kullanıcılara bir sıra vermekte ve etkileşimi en üst sıradakinden başlayarak yapmaktadır. Bu sayede kötü niyetli olan kullanıcılar zamanla izole edilerek etkileşime girmeleri engellenmektedir.

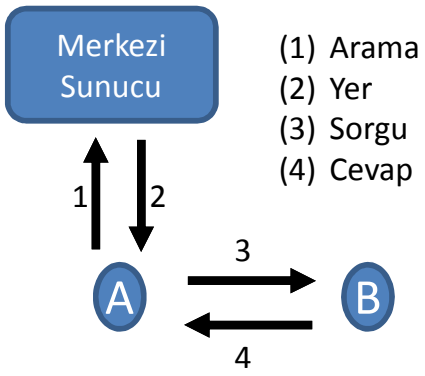
Tez içeriği genel olarak şu şekildedir: Bölüm 2'de eşler arası sistemlerin genel yapısı ve türleri anlatılmıştır. Bölüm 3'te itibar ve güven tanımlanarak eşler arası sistemlerdeki güven yönetimi, türleri ve bu kapsamda yapılmış çalışmalardan bahsedilmiştir. Bölüm 4'te makine öğrenmesi yaklaşımları kısaca anlatılmıştır. Bununla birlikte eşler arası sistemlerdeki güven yönetimini sağlamak adına kullanılan makine öğrenmesi teknikleri ve çalışmaları özetlenmiştir. Bölüm 5'te, tez kapsamında geliştirilen güven yönetim modeli yapısı ve modüllerin işleyişleri anlatılmıştır. Bölüm 6'da modelin eğitimleri ve eğitim sonuçlarına göre yapılan testler ve deneyler sonuçları ile birlikte anlatılmıştır. Bölüm 7'de ise yapılan çalışmaların ve modelin davranışlarının özeti anlatılarak sonuçlandırılmıştır.

2 EŞLER ARASI SİSTEMLER

Eşler arası sistemler, genel olarak uygulama düzeyinde bilgisayarlar arasında kaynak paylaşımını sağlayan sistemlerdir. Bu sistemlerin temeli istemci/sunucu mimarisine dayanmaktadır. Fakat bu mimariden en temel farkları, kullanıcıların aynı anda hem istemci ve hem de sunucu olarak davranmalarıdır. Eşler arası sistemlerin genel çalışma mekanizması, bir eşin ihtiyaç duyduğu bir hizmete, sisteme dâhil olmuş diğer eşler üzerinden ulaşması şeklindedir. Bu sistemler sayesinde sınırsız kaynak erişimi sağlanmaktadır. Eşler arası sistemler yapısal olarak merkezi, yarı merkezi ve dağıtık olarak üç ayrı sınıfa ayrılabilir.

2.1 Eşler Arası Merkezi Sistemler

Merkezi sisteme dayalı eşler arası sistemler, bilgilerin yönetimini ve kullanıcılar arasındaki iletişimi merkezi bir sunucu kullanarak sağlamaktadırlar. Genel çalışma prensibi olarak kullanıcılar kendilerine yakın bir sunucuya bağlanıp ellerindeki kaynak listesini sunucudaki veritabanına eklerler. Başka bir kullanıcı bir kaynağı/bilgiyi aradığında merkezi sunucu üzerindeki veritabanına erişir ve ilgili kaynağı/bilgiyi paylaşan kullanıcı (eş) bilgisine erişir. Bu kullanıma en güzel örnek olarak Napster [11] verilebilir. Sunucular üzerinde şarkı paylaşan kullanıcıların listesini tutan Napster, merkezi bir yapı içermektedir. Napster gibi merkezi sistemlerin en temel problemi, sunucular üzerinde oluşan bir hatadan sistemin tamamının etkilenir konumda olmasıdır. Şekil 1 merkezi otoriteye sahip eşler arası bir sistemin çalışmasını özetlemektedir.



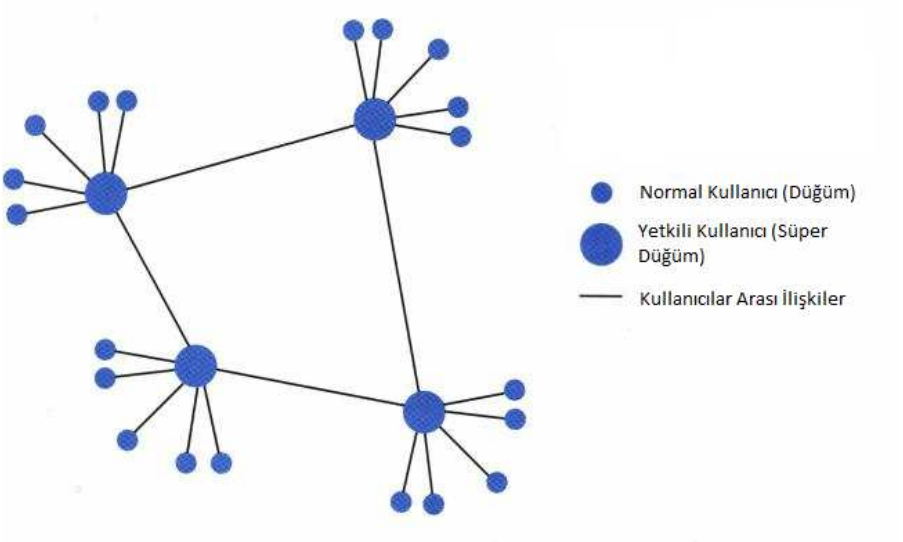
Şekil 1: Merkezi Sistem Çalışma Mantığı

2.2 Eşler Arası Yarı Merkezi Sistemler

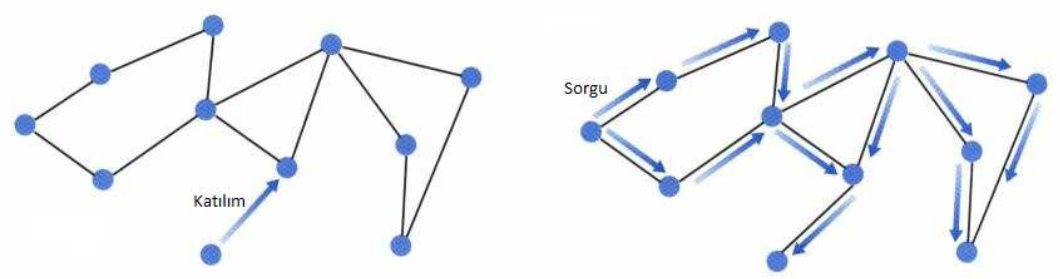
Bu sistemler özelliklerinin bir kısmını merkezi, bir kısmını da dağıtık eşler arası sistemlerden almaktadırlar. Yarı merkezi sistemlerde yetkilendirilmiş ve diğer kullanıcılardan farklı olan kullanıcılar süper düğümler (supernodes) olarak adlandırılır. Süper düğümler daha fazla kapasiteye sahip ve daha fazla sorumluluk alan kullanıcılar olarak tanımlanabilir. KaZaA [12] bu sistemlere örnek olarak verilebilir. KaZaA merkezi yapıdan farklı olarak ayrıcalık tanıdığı bir kullanıcıya süper düğüm görevi vermektedir. Merkezi ve özelleştirilmiş bir sunucudan farklı olarak bu süper düğüm hem kullanıcı hem de diğer normal kullanıcıların bilgilerini tutan bir sunucu mantığında çalışmaktadır. Eşler arası sistemlerin büyüklüğünden dolayı oluşan yükü KaZaA, yaptığı ikili kullanıcı sınıflandırması (normal düğümler ve süper düğümler) ile hafifletmektedir. Şekil 2 KaZaa ağının yapısını göstermektedir. Skype, Morpheus, Gnutella2 gibi sistemler de yine eşler arası yarı merkezi sistemlere örnek verilebilir.

2.3 Yapısal Olmayan Eşler Arası Dağıtık Sistemler

Dağıtık sistemler merkezi bir otorite ya da sunucu yapısı olmadan çalışan sistemlerdir. Bu tip sistemlerde tüm kullanıcılar aynı rolü oynarlar. Dağıtık sistemlerde kaynak arama maliyeti yüksek olurken kullanıcılarda



Şekil 2: Kazaa Ağı [1]



Şekil 3: Gnutella Ağı [1]

oluşacak hataların sisteme etkisi çok düşük olmaktadır. Belirli kurallar çerçevesinde sorgular ağa yayılmakta ve gelen cevaplara göre istenen bilgiye ulaşılmaktadır. Fakat ağa gönderilen sorgular bant genişliğine ve yoğunluğuna göre ağ trafiğini arttırabilmektedir.

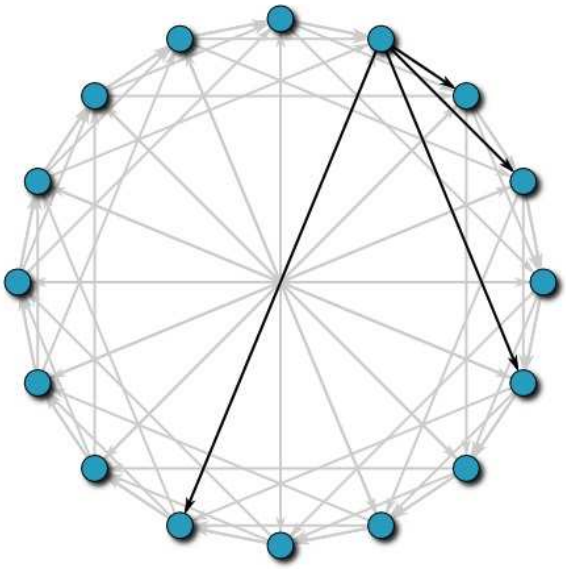
Dağıtık sistemler de genel olarak iki başlık altında incelenebilir. Bunlardan ilki yapısal olmayan (*unstructured*) eşler arası dağıtık sistemlerdir. Bu sistemlere en iyi örneklerden birini Gnutella [5] temsil etmektedir. Gnutella, diğer birçok eşler arası sistem gibi kendi yönlendirme mekanizmasını oluşturmaktadır. Merkezi bir yapı olmayan Gnutella'da kullanıcılar doğrudan birbirine bağlanmakta ve sistemde istemci ya da sunucu olarak görev almaktadırlar. Yapısal olmayan diğer sistemlerde olduğu gibi Gnutella'da da bir kaynağın nerede olduğu bilinmediği için deterministik olmayan

arama mekanizması kullanılmaktadır. Bir kaynak araması sırasında sorgu, komşular aracılığı ile ağa yayılmakta ve kaynak aranmaktadır. Sorgu yaşam süresi (*Time-to-Live TTL*) sonlanana kadar sorgu görevini yerine getirmektedir [13]. Şekil 3 Gnutella genel yapısını göstermektedir. Freenet [14, 11] de yapısal olmayan sistemlere Gnutella benzeri bir yapı ile örnek teşkil etmektedir. Freenet kullanıcıları veri depolama aygıtlarının kullanmadıkları kısımlarını paylaşıma açmaktadırlar. Açılan bu alan ağ için okuma ve yazmaya uygun hale getirilmektedir. Her bir kullanıcı, kendilerinin olmayan bilgileri içerebilecek yerel bir veri depolama alanı yönetir. Her bir kaynak biricik ve yer bağımsız anahtar ile temsil edilmektedir. Kullanıcılar ellerindeki kaynakların anahtarlarını ve o anahtar ile temsil edilen kaynakları içeren kullanıcıların bir listesini tutarlar. Yönlendirme işlemlerini kendileri yönetirler ve bu listeler sayesinde yönlendirmeleri yaparlar. Bir dosyayı arama yöntemi olarak ise derinlik öncelikli arama (*depth first search* [15]) tekniğini kullanırlar.

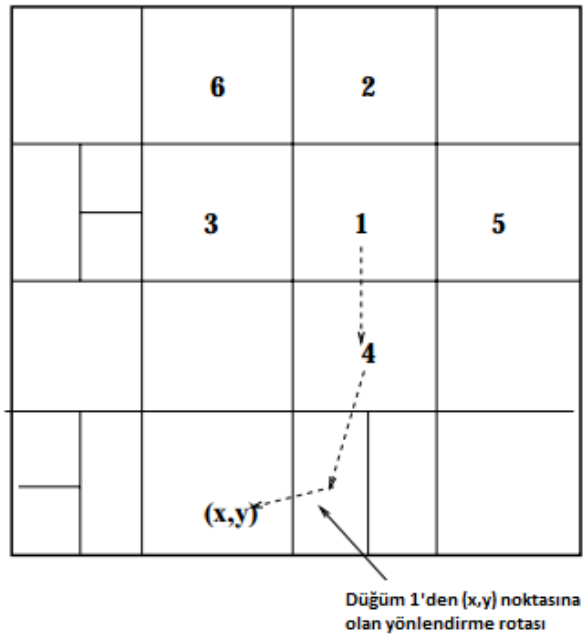
2.4 Yapısal Eşler Arası Dağıtık Sistemler

Dağıtık sistemlerde ikinci başlık ise yapısal eşler arası dağıtık sistemlerdir. Chord [16] verimli eş ve kaynak tahsisi sağlayan dağıtık bir arama protokolü olarak bu tip sistemlere örnek teşkil eder. Chord temel olarak tek bir işlev gerçekleştirmektedir. Bu işlev ile bir anahtar tahsis etmekte ve bu anahtar ile ilgili bir kaynağı barındıran bir kullanıcıyı temsil etmektedir. Anahtarların kullanıcıları temsil etmesi için tutarlı özetleme (*consistent hashing* [17]) yöntemi kullanılmaktadır. Bu yöntem sayesinde bir kullanıcı üzerindeki anahtar sayısının dengeli olarak dağıtılması sağlanır. Aramalar bu anahtarlar üzerinden yapılmakta ve sonuçlara göre kaynaklara ulaşılmaktadır. Şekil 4 Chord'un genel yapısını göstermektedir.

CAN [2] de yapısal dağıtık sistemlerden biridir. Temel mimari olarak Dağıtık Özet Çizelgesi kullanmaktadır. Anahtar ve veri eşlemesini sağlayarak



Şekil 4: Chord Ağı



Şekil 5: CAN Yapısı[2]

indeksleme ve ölçeklendirilebilir bir yönlendirme kabiliyeti sağmaktadır. Şekil 5 bu sistemin genel akışını göstermektedir. CAN, yapısal tasarım olarak sanal çok boyutlu kartezyen koordinat alanından oluşmaktadır. Pastry [18] ise çok büyük alana sahip eşler arası sistemlerde yer bulma ve yönlendirme işlemlerini yapabilen dağıtık sistemlerden biridir. Tamamen dağıtık çalışan Pastry, aranılan bilgileri bir sonraki arama ihtimaline karşı yakın kullanıcılarda tutarak yönlendirme maliyetini düşürmektedir. Bir yönlendirme ya da arama işleminde ağın yapısını da hesaba katarak ulaşılacak kaynağa en kısa yoldan gitmeye çalışmaktadır. Tapestry [19] bir başka eşler arası dağıtık sistemlerden biridir. Tıpkı Pastry'da olduğu gibi ağ üzerindeki sorguların ölçeklendirilebilir ve etkili bir şekilde yönlendirilmesini sağlamaktadır. Yönlendirme ve bilgiye ulaşım maliyetlerini düşüren Tapestry, başarılı bir eşler arası ağ oluşturulmasını sağlamaktadır.

3 İTİBAR VE GÜVEN YÖNETİMİ

3.1 Güven

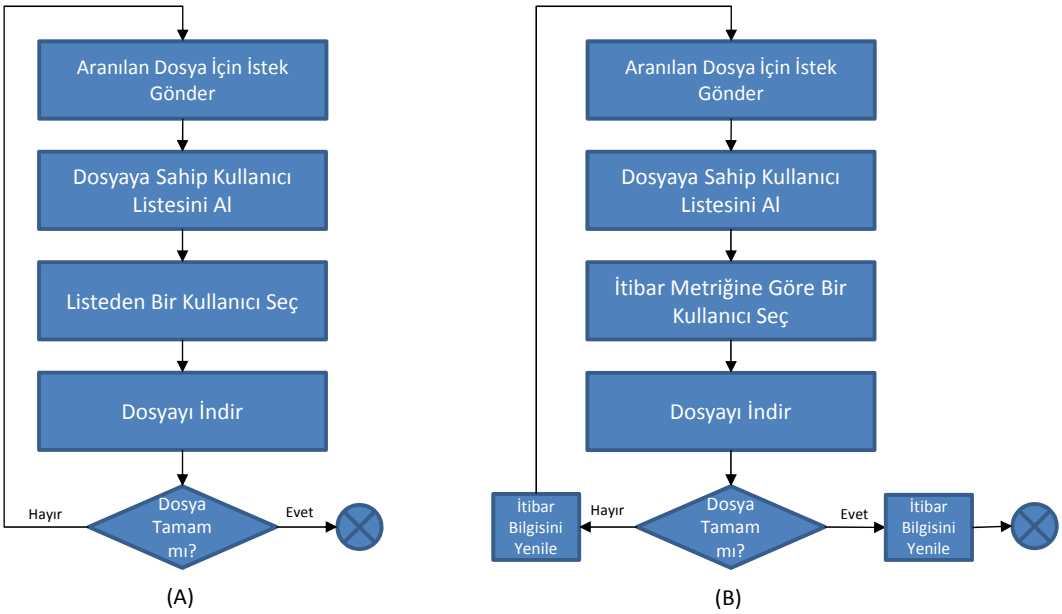
Güven, insanların varoluş dönemlerinden beri önemli olan bir kavramdır. İnsanlar topluluk olarak yaşamaya başladıklarından beri birbirleri ile iletişimlerinde güven ihtiyacı hissetmişlerdir. İnsanlar arasında önemli olan bu kavramı açık ve tam bir şekilde tanımlamak zor bir süreçtir.

Psikoloji, sosyoloji, tarih, hukuk, ekonomi gibi alanlarda araştırma yapan insanlar, farklı konseptlerde ve bakış açılarında güven tanımı yapmaya çalışmışlardır. Oxford sözlüğüne göre güven, doğruluk, yetenek ya da güç gibi unsurlarda birisine ya da bir şeye duyulan sağlam bir inançtır. 1973 yılında Deutsch'un [20] yaptığı tanımlamaya göre güven, bir bireyin korkulan yerine başka birinden arzulanan şeyi bulmasına olan eminliğidir.

Grandison ve Sloman'ın [21] 2000 yılında yaptığı tanımlamaya göre güven, tutarlı ve bağımlı şekilde hareket eden bir varlığın yetkinlik inancıdır. Bizim ilgilendiğimiz alana göre güven tanımı 2005 yılında Chang [22] tarafından yapılmıştır. Bu tanıma göre güven; güvenilen kullanıcının verilen içerikte ve verilen zaman aralığında karşılıklı olarak kabul hizmeti sunmak için yetenekli ve güvenilir kullanıcı istekliliğine sahip olmasına olan inançtır.

3.2 İtibar

İtibar kavramı farklı disiplinlerde kullanılan geniş kapsamlı bir tanımı ifade etmektedir. Oxford sözlüğüne göre itibar, birisinin ya da bir şeyin hakkında bilinen genel inançlar ve görüşlerdir. Abdul Rahman'a [23] göre itibar tanımı, bir kullanıcının geçmiş deneyimleri hakkında sahip olunan bilgiye dayanarak öngörülen beklentiler şeklindedir. Sabater



Şekil 6: (A) Geleneksel Eşler Arası Sistemlerdeki Yaşam Döngüsü, (B) İtibar Tabanlı Eşler Arası Sistemlerdeki Yaşam Döngüsü

[24] ise bu kavramı bir şey hakkındaki görüş ya da bakış açısı olarak tanımlamaktadır. Mui [25] tarafından yapılan çalışmada itibar, bir kullanıcının niyetleri ve normları doğrultusunda geçmiş davranışlarına dayanarak diğer kullanıcıların gözünde yarattığı bakış açısıdır.

Tüm bu tanımlardan yola çıkarak itibar tanımını şöyle yapılabilir: Bir kullanıcının geçmişte yapmış olduğu hareketlerini, diğer kullanıcıların yeni bir hareketten önce karar unsuru olarak kullanabilmek için yaptıkları değerlendirme o kullanıcının itibarını belirlemektedir.

3.3 Eşler Arası İtibar Tabanlı Güven Yönetim Sistemleri

Eşler arası sistemlerin günümüzdeki yaygınlığı ile beraber güvenlik sorunları da giderek artmaktadır. Bu sorunların çözümü için itibar tabanlı güven yönetim sistemleri önerilmiştir. Normal bir eşler arası sistem ile itibar tabanlı sistemler arasında temel farklılıklar bulunmaktadır. Şekil 6 ikisinin de genel yaşam döngüsünü göstermektedir.

Normal sistemlerde aranan kaynağı içeren kullanıcıların herhangi birinden ya da bir kaçından kaynak alınırken, itibar tabanlı sistemlerde kaynağı içeren kullanıcılardan itibar metriğine göre yüksek değere sahip olanlar tercih edilir. İtibar metriğine göre yapılan işlemin ardından da etkileşime girilen kullanıcının itibar metriği tekrar güncellenir. Bu şekilde devam eden döngü sayesinde kötü amaçlı kullanıcılar zaman içerisinde sistemde aktif olamayacak hale gelirler.

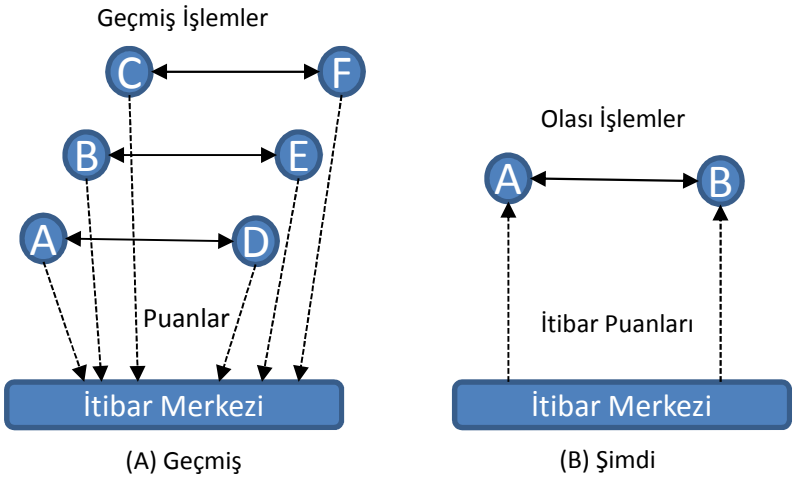
İtibar tabanlı sistemler, Resnick'e [26] göre genel olarak üç özelliği göz önüne almalıdırlar. Bu üç özellik aşağıda verilmiştir:

- Varlıklar uzun ömürlü olmalıdırlar. Bu yüzden her bir etkileşim, gelecek etkileşimler için beklenti oluşturmaktadır.
- Var olan etkileşimler için verilen derecelendirmeler saklanabilmeli ve dağıtılabilmelidir.
- Geçmiş etkileşimlere dayanan derecelendirmeler gelecek etkileşimlere karar vermede yol gösterici olmalıdır.

İtibar tabanlı güven yönetim sistemleri, sistemin temel mimarisine göre genel olarak iki başlık altında sınıflandırılabilir. Bunlar merkezileştirilmiş itibar sistemleri ve dağıtık itibar sistemleridir.

3.3.1 Merkezileştirilmiş İtibar Sistemleri

Merkezileştirilmiş itibar sistemlerinde, bir kullanıcının performansı ve davranış eğilimi, bu kullanıcı ile doğrudan etkileşime girmiş sisteme üye diğer kullanıcılar ile yaptıkları etkileşimlerin kalitesine göre derece ve görüş olarak toplanmaktadır. Yapılan etkileşimlere göre toplanan derece ve görüşler merkezi bir otoriteye, başka bir deyişle itibar merkezine gönderilmektedir. Gönderilen tüm derece ve görüşler merkez sayesinde

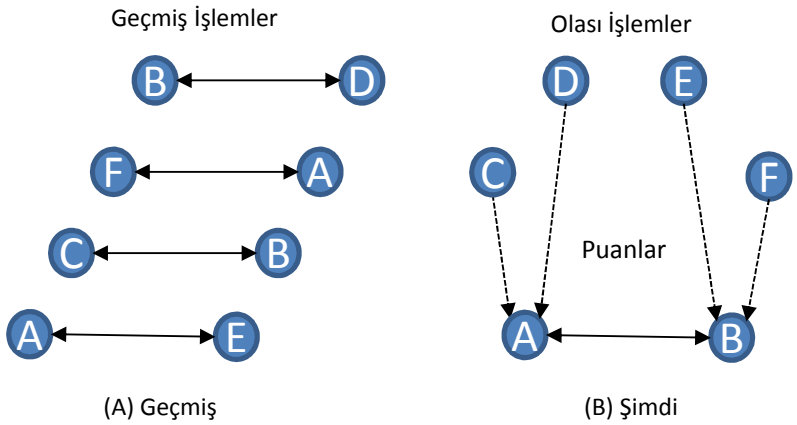


Şekil 7: Merkezileştirilmiş İtibar Sisteminin Genel Mimarisi

itibar puanına çevrilip kamuya açık hale getirilmektedir. Bir kullanıcı başka bir kullanıcı ile etkileşime girmeden önce kullanıcının itibar puanına bakmakta ve etkileşime girip girmeyeceğine karar vermektedir.

Şekil 7 merkezileştirilmiş bir itibar sisteminin genel işleyiş şemasını göstermektedir. A geçmişte yaşanan etkileşimleri, B ise yeni bir etkileşim öncesindeki durumu göstermektedir.

Yapılan her bir etkileşimin ardından kullanıcılar birbirlerinin performansını puanlandırmaktadır. Verilen performans puanları itibar merkezi tarafından toplanmakta ve bu kullanıcılar hakkındaki global itibar puanını oluşturmaktadır. İtibar merkezi bu işlemi düzenli olarak her etkileşim sonrasında gerçekleştirmektedir. Sürekli güncel tutulan itibar puanları tüm kullanıcıların göreceği şekilde yayınlanmaktadır. Bu sayede kullanıcılar başka bir kullanıcı ile etkileşime girmeden önce, o kullanıcı hakkındaki geçmiş tecrübelere bakıp karar verebilmektedir. Şekil 7-A'da görüldüğü üzere A ve B kullanıcıları D ve E kullanıcıları ile etkileşime girmekte ve etkileşim sonundaki puanlarını itibar merkezine göndermektedirler. Daha sonra Şekil 7-B'de de görüldüğü gibi A ve B kullanıcıları kendi aralarında etkileşime girmeden önce itibar merkezinden birbirleri hakkındaki puan bilgisini almakta ve ona göre etkileşim kararı vermektedirler.



Şekil 8: Dağıtık İtibar Sisteminin Genel Mimarisi

3.3.2 Dağıtık İtibar Sistemleri

Dağıtık itibar sistemleri, eşler arası sistemlerin temel yapısına daha uygun çalışma mantığına sahiptirler. Dağıtık sistemlerde derecelendirmelerin toplandığı ya da itibar puanının hesaplandığı merkezi bir otorite bulunmamaktadır. Bunun yerine, basit bir şekilde kullanıcılar her bir kullanıcı ile yaptığı etkileşimlere ait derecelendirme ve itibar puanlarını kendileri saklamakta ve istenildiği zaman bu bilgileri diğer kullanıcılar ile paylaşmaktadırlar. Bir kullanıcı başka bir kullanıcı ile etkileşime başlamadan önce, o kullanıcı ile önceden etkileşime girmiş mümkün olduğunca fazla sayıda kullanıcıdan bilgi toplamakta ve bu bilgiler ile kendi derecelendirmelerini birleştirerek bir karar vermeye çalışmaktadır. Şekil 8 dağıtık bir itibar sisteminin genel işleyişini göstermektedir.

Bu tip sistemlerde hem itibar puanının hesaplanması hem de hesaplanmış puanların kullanıcılar arasında dağıtılması büyük önem taşımaktadır. Kullanıcılar hem kendi geçmiş deneyimlerini saklamakta hem de diğer kullanıcıların geçmiş deneyimlerinden faydalanmaktadırlar. Fakat burada da geçmiş deneyimlerini aldıkları kullanıcıların ne kadar güvenilir oldukları sorunu ile karşılaşılmaktadır. Deneyimlerini paylaşan kullanıcılardan bazıları kötü niyetlerle yanıltıcı bilgiler verebilmekte ve böylece sistemin işleyişine zarar verebilmektedirler. Bu durum özellikle eşler arası

sistemlerde güven yönetimini zorlaştırmaktadır.

3.4 İtibar Tabanlı Güven Yönetim Modelleri İle İlgili Çalışmalar

Rahman [27] tarafından 1997 yılında tavsiye protokolüne dayalı dağıtık çalışan güven yönetim modeli ortaya atılmıştır. Modelin odaklandığı dört unsur bulunmaktadır; dağıtık, genelleştirilmiş, anlamsal güven ve tavsiyeler.

Dağıtık yapı, her bir kullanıcının kendi güven kurallarından sorumlu olmasını ve diğer kullanıcılarla ilişkilerinde bunları kendilerinin yönetmesini sağlamaktadır. Genelleştirilmiş bir güven anlayışı, farklı güven sınıflandırmalarının ve farklı karakterlerin olduğu bir sistemde sınırların bilinmesine ve genel bir görüş oluşturulmasına yardımcı olmaktadır. Anlamsal güven, güven değerlerinin karşılaştırılabilir olmasını sağlamaktadır. Son olarak tavsiyeler sayesinde kullanıcı sistem içerisinde tekil olmaktansa tanıdığı diğer kullanıcılardan alacağı tavsiyeler sayesinde tanımadıkları hakkında bilgi sahibi olmaktadırlar.

Önerilen bu modelde güven ilişkileri tamamen iki varlık arasında kurulmaktadır. Karşılıklı güven iki ayrı güven ilişkisi ile temsil edilmektedir. Bu iki ayrı güven ilişkisi birbirinden farklı özellikler ile ayırt edilebilir. Bir kullanıcı başka bir kullanıcıya güveniyorsa bu doğrudan güven ilişkisini temsil etmektedir. Fakat bir kullanıcı başka bir kullanıcıya diğer kullanıcılardan aldığı tavsiyeler doğrultusunda güveniyorsa aralarındaki ilişki tavsiye tabanlı güven ilişkisi olarak ifade edilir[28]. Önerilen model, merkezi bir otoriteyi desteklemediği için güven ilişkileri sadece her bir kullanıcının kendi veritabanında tutulmaktadır. Bahsedilen iki tür güven ilişkisine karşılık olarak da her bir kullanıcı tarafından iki tür veri yapısı idame edilmektedir. Veri yapılarından ilki doğrudan edinilen güven deneyimlerinden, ikincisi ise diğer kullanıcılardan alınan tavsiye deneyimlerinden oluşmaktadır.

Tavsiyeler sadece doğrudan güven ilişkisi olmadığı zamanlarda işlem için kullanılmak adına hesaba katılmaktadır.

İtibar kavramı bu model için kullanıcı adı, güven kategorisi ve özel güven değerinden oluşmaktadır. Dağıtık mimarinin gereği olarak her bir kullanıcı kendi güven değerlerini depolamaktadır. Aynı zamanda her bir kullanıcı anahtar tabanlı şifreleme tekniği ile mesajlarını göndermekte ve kötü niyetli kullanıcıların mesajlara müdahalesini engellemeye çalışmaktadır. Fakat kullanıcıların kimlik koruması adına bir önlem bulunmamaktadır. Bu da sistemin güvenirliliğini düşürmektedir. Her bir kullanıcı, tavsiyeleri değerlendirmekte ve doğrudan güven değeri ile tavsiye tabanlı güven değerlerini de hesaba katarak itibar puanını hesaplamaktadır. Güvenilen bir kullanıcı, doğrudan güven değerine sahip ise aynı kullanıcı için tavsiyeleri hesaba katmamaktadır. Depolama maliyeti kullanıcının etkileşim sayısına ve aldığı tavsiye miktarına göre değişmektedir. Bu da fazla kullanıcı ile etkileşime girmiş bir kullanıcı için yüksek depolama maliyeti anlamına gelmektedir. Ayrıca bu model, kullanıcının bağlantısının kopması ya da aniden çıkması gibi durumları ele almamaktadır. Bu da bütünlüğü etkileyecek bir unsur teşkil etmektedir.

BinaryTrust [3] ikili güven tabanına dayanmaktadır. Bir kullanıcı ya güvenilir ya da güvenilmez olarak nitelendirilir. Kullanıcılar arasındaki her bir etkileşim de ya doğru olarak ya da hatalı olarak sınıflandırılır. Bir kullanıcı kötü bir davranış sergilediğinde güvenilmez olarak işaretlenir ve şikâyet diğer tüm kullanıcılara gönderilir. Modelde kötü niyetli bir durum, aykırı bir hareket olduğundan dolayı sadece dürüst olmayan etkileşimler hesaba katılmaktadır. Bir kullanıcının itibarı sistemdeki genel şikâyet bilgileri üzerinden hesaplanmaktadır. Veri saklama yapısı olarak kullanılan PGrid, kullanıcı şikâyetlerini depolamaktadır. Bu güven modeli genel olarak aşağıdaki gibi çalışmaktadır;

- Bir kullanıcı başka bir kullanıcı hakkındaki şikayetini oluşturur ve bunu

insert Messages yöntemiyle diğer kullanıcılara gönderir.

- Bir kullanıcı başka bir kullanıcının güvenilirliğini sorgulamak istediğinde, o kullanıcı hakkındaki şikayetleri arar. Ağ üzerindeki trafiği azaltmak adına, belirli bir sayıda kullanıcıdan benzer güven bilgilerini aldıktan sonra aramayı sonlandırır ve aldığı bilgilere göre etkileşim yapıp yapmayacağına karar verir.

Şikayet mekanizmasını temel alan BinaryTrust'ın bazı dezavantajları bulunmaktadır. Bunlar;

- Kullanıcılar sisteme yeni bir kimlikle kaydolarak hakkındaki şikayetleri silebilmektedirler.
- Bazı kullanıcılar kendileri hakkındaki şikayetleri alabilirler. Aldıkları şikayetleri kendi güven değerlerini arttırabilmek adına da silebilirler.
- Keyfi ve kötü niyetli şikayetleri engellemek adına bir mekanizma bulunmamaktadır.
- PGrid kullanım yapısının iyileştirilmesi gerekmektedir.

Genel mantığı dağıtık oylama algoritması olan P2PRep [4] bir kaynağa erişmek isteyen kullanıcının diğer kaynağı sağlayan kullanıcılar hakkındaki itibarı öğrenmek adına kullanıcıları yoklama (*polling*) yöntemini kullanmaktadır. Kullanıcı, kaynak sahiplerini öğrenmek adına attığı sorguya aldığı cevap neticesinde, kaynağa sahip kullanıcılardan bir küme seçerek onlar hakkındaki itibarı öğrenmek için diğer kullanıcıların fikirlerini almaktadır. Algoritma iki tür oylama yöntemini sağlamaktadır; basit oylama (*basic polling*) ve gelişmiş oylama (*enhanced polling*). Basit oylamada diğer kullanıcılar kaynak sahibi hakkındaki fikirlerini oylayarak göndermekte ve sorguda bulunan kullanıcı bu oylardan en güvenilir kaynak sahibine karar vermektedir. Gelişmiş oylamada ise fikirlerini belirten ve oylama yapan

kullanıcılar aynı zamanda kimlik bilgilerini de göndermektedir. Kimlik bilgisi, kullanıcıların oylarını ağırlıklandırmada kullanılmaktadır.

İnanılırlık yönetimi (*Credibility Management*) gelişmiş oylamada kullanılmaktadır. Kaynak sağlayıcısı hakkındaki oyunu gönderen kullanıcının güven değeri, alınan oylar ile itibar değerini hesaplarken göz önüne alınmaktadır. İnanılan kullanıcılar bu hesaplamada daha fazla ağırlık almaktadırlar ve kararlarda daha etkin rol oynamaktadırlar. Fakat basit oylamada inanılan kullanıcılar dikkate alınmamaktadır.

Önerilen model, oylanan kullanıcılar ve onların oyları nedeniyle önemli bir ek yük getirmektedir. Basit oylama metodunda kullanıcının oyu sağlayan olup olmadığı *TrueVote* ve *TrueVoteReply* mesajları ile kontrol edilmektedir. Geliştirilmiş oylama metodunda ise oylayıcının kimliğini doğrulamak adına *AreYou* ve *AreYouReply* mesajları gönderilmektedir. Bu da ağ yükünü arttırmaktadır.

XRep [29] protokolünde kullanıcılar ve kaynakların beraber ele alınarak itibar puanı hesaplanmaktadır. XRep protokolünün itibarın idamesi ve değişimi için kullanıldığı ve bunun eşler arası sistemlerde var olan güvenlik saldırılarına karşı avantaj sağladığı belirtilmektedir [29]. XRep protokolü her bir kullanıcının kendi deneyimleri ve kaynak bilgileri ile kendi bilgilerini yönetmekte ve diğer kullanıcılar ile bu bilgilerin paylaşılmasını sağlamaktadır. Her bir kullanıcı iki deneyim deposunu (*experience repositories*) yönetmektedir;

- **Kaynak Deposu (*Resource Repository*):** Her bir kaynağın kimlik bilgisi ile kaynağın iyi ya da kötü olduğunu bildiren ikili değeri tutan depodur.
- **Kullanıcı Deposu (*Servent Repository*):** Kullanıcıların kimlik bilgileri ile birlikte başarılı ve başarısız kaynak indirme sayısını tutan depodur.

XRep protokolü beş aşama ile işlem yapmaktadır. Bunlardan ilki kaynak

arama aşamasıdır. Bu aşamada aranılan kaynağa sahip kullanıcılar bulunmaya çalışılır. İkinci aşamada gelen sonuçlar arasından kaynak seçilmektedir. Seçilen kaynak için kaynak istemcisi olan kullanıcı tarafından, kaynak ve kaynak sağlayıcı kullanıcılar hakkındaki itibar için oylama sorgusu gönderilir. Gelen oy sonuçlarına göre istenilen kaynağın itibar puanı üçüncü aşama olarak hesaplanır. Dördüncü aşamada ise kaynağı sağlayan en iyi kullanıcı kontrol edilir ve karar verilir. Son aşamada ise karar işleminin ardından seçilen kullanıcı ile doğrudan kaynak indirme yapılır.

XRep'in kaynak ve kaynak sağlayıcı kullanıcılar hakkında itibar puanı hesaplamak için yaptığı sorgular ağ trafiğine fazladan yük getirmektedir. Bu yük Gnutella ağı yapısından kaynaklanmaktadır. Ayrıca hem kaynak hem de kaynak sağlayıcılar üzerine yapılan itibar yönetim sistemi modelinin performansının her iki açıdan da kanıtlanması adına gereken sonuçlar bulunmamaktadır. Sadece kaynak sağlayıcı kullanıcılar üzerine itibar görülmektedir.

EigenTrust [9] algoritması sistemdeki her bir kullanıcı için kullanıcının geçmişindeki hareketlere göre global bir güven değeri atamaktadır. Bu güven değeri kullanıcının diğer kullanıcılar ile olan deneyimlerini yansıtmaktadır. Yazarlar, küresel güven değerini hesaplayabilmek adına *power iteration* tekniğini kullanan dağıtık ve güvenli bir yöntem önermektedirler.

EigenTrust başlıca aşağıdaki karakteristikleri içermektedir;

- **İtibar verisi (*Reputation Data*):** Yerel güven değerinin temsil ettiği memnun olunan ya da olunmayan etkileşim sayısı. Yerel güven değeri, 0 ve 1 arasında normalize edilmektedir.
- **İtibar Hesaplaması (*Reputation Computation*):** EigenTrust yapısı geçişli güven (*transitive trust*) değerine dayanmaktadır. Kullanıcının diğer kullanıcılar tarafından verilen yerel güven değerleri ile

hesaplanan küresel güven değeri diğer atanan kullanıcıların küresel güven değerlerine göre ağırlıklandırılmaktadır. Küresel güven değeri, bir matrisin başlıca eigen vektörünün normalize edilmiş yerel güven değerine karşılık gelmektedir. Her bir kullanıcı kendi itibar puanını kendisi hesaplayıp raporladığı için eğer kötü niyetli bir kullanıcı ise kolaylıkla sahte itibar puanı raporlayabilecektir. Bu yüzden güvenli EigenTrust yapısında küresel güven değerini hesaplamak adına atanmış puan yöneticileri (*score managers*) bulunmaktadır ve bunlar *distributed hash table* kullanmaktadırlar. Her bir puan yöneticileri bir grup kullanıcıdan sorumludur. Yöneticiler her bir kullanıcı için ilgili kullanıcıdan indirilen dosya bilgilerini ve güven atamalarını öğrenmektedir.

- **İnanılırlık Mekanizması (*Credibility Mechanism*):** Bir kullanıcının küresel güven değerini hesaplamak adına farklı puan yöneticileri kullanılır. Kötü niyetli puan yöneticilerinin de sistemde var olabileceği göz önüne alınırsa, oy çoğunluğu mantığı uygulandığı için kötü niyetli puan yöneticilerinin sahte güven değeri vermelerinin etkisi en aza indirilebilmektedir.

- **Kötü Niyetli Kullanıcı Politikası (*Malicious Peer Policy*):** EigenTrust sisteme zarar vermeye çalışan kötü niyetli kullanıcı gruplarına karşı dirençli bir yapıdadır. Güvenli EigenTrust yapısında tek taraflı özet fonksiyonu (*one-way hash function*) kullanılmaktadır. Bu sayede puan yöneticilerinin kimin küresel güven değerini hesapladığını bilmesi mümkün değildir. Kötü niyetli kullanıcılar birbirlerinin itibar puanlarını arttıramazlar. Ek olarak, kullanıcılar kendi güven bilgilerini saklayan kullanıcıları bilmediklerinden kendi güven değerlerini manipüle edemezler.

EigenTrust algoritması etkili bir çözüm sunmaya çalışmasının yanı sıra bazı ek yükler getirmektedir. Her bir kullanıcı, küresel güven değerini toplamak

ve hesaplayabilmek için uzun zaman harcayabilmektedir. Ayrıca her bir kullanıcının güven değerini toplayabilmek ve hesaplayabilmek adına dağıtık özet tablosu (*distributed hash table*) ve puan yöneticileri kullanımı iletişim maliyetine ek maliyet getirmekte ve ağ trafiğini artırmaktadır.

4 MAKİNE ÖĞRENMESİ İLE GÜVEN YÖNETİMİ

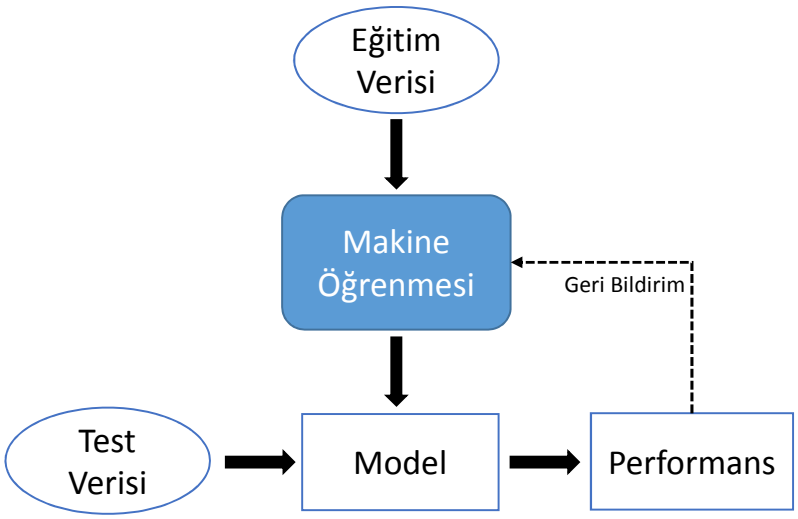
Bu bölümde genel olarak makine öğrenmesinden ve makine öğrenmesinin eşler arası sistemlerde güven yönetimindeki uygulamalarından bahsedilecektir. Ayrıca makine öğrenmesi tekniklerinden biri olan genetik programlama hakkında giriş düzeyinde bilgi verilecektir.

4.1 Makine Öğrenmesi

Yazılım alanında bir problemin çözümü için genelde uygun bir algoritmaya ihtiyaç duyulmaktadır. Algoritmanın adımları sayesinde girdiler uygun çözümü oluşturan çıktılara dönüşmektedir. Örneğin sıralama algoritmasında sıralanması gereken sayılar bir liste olarak alınır ve çıktı olarak da sıralı halleri verilir. Fakat karmaşık problemler için etkin ve etkili algoritma bulunması zor bir problemdir. Bazı problemlerin girdileri belirli olsa bile çıktıları birçok faktöre bağlı olabilmekte ve geniş bir çözüm kümesine sahip olabilmektedir. Büyük miktardaki bu verilerin elle işlenmesi ve analizinin yapılması ise mümkün değildir.

Günümüz teknolojileri sayesinde büyük miktarlardaki ham veriler saklanabilmektedir. Gerektiğinde saklanmış bu veriler ile sonuçlara ulaşmak ve analizler yapmak mümkündür. Problemlere aranılan çözümler bu veriler içerisinden bulunmaktadır. Çünkü verilerin belirli karakteristikleri vardır. Örneğin bir markete giden insanların alışveriş alışkanlıkları, yaptıkları alışverişin sırasına ve miktarına göre belirlenebilmektedir. Marketin içindeki ürünlerin dizilimleri bile, müşterilerin alışveriş verilerinden yapılan çıkarım ile düzenlenebilmektedir.

Kesin çözümler her türlü problem için mümkün olmayabilir. Bu sebepten dolayı çıkarsama ile çözüme yönelik bir yöntem daha doğru sonuç verebilmektedir. Geçmişteki deneyimlere dayanarak gelecekteki sonuçları



Şekil 9: Makine Öğrenmesi

tahmin etmek için kullanılan yönteme Makine Öğrenmesi (*Machine Learning*) denilmektedir. Makine öğrenmesi veri madenciliği, bilgisayarlı görü, dil işleme gibi alanlarda kullanılabilmektedir [30, 31]. Ham verilerden çıkarsama yapma ve çözüm uzayında etkili bir çözüm bulmak için veri madenciliği kullanılmaktadır. Bunun gibi daha birçok alanda uygulamaları görülmektedir.

Makine Öğrenmesi genel olarak iki adımda sonuç vermektedir. İlk olarak öğrenme fazı bulunmaktadır. Bu aşamada büyük miktardaki verilerden istenilen çözüme dair kabul edilebilir bir şablon çıkarılmaya çalışılmaktadır. Ardından ikinci adım olarak çıkarılan şablon yeni veriler üzerinde denenmekte ve sonuçların doğruluklarına göre kabul görmektedir [32]. Şekil 9 makine öğrenmesinin genel işleyişini göstermektedir.

Makine Öğrenmesi yönteminde farklı uygulamaların, analizlerden farklı beklentileri olmaktadır. Makine öğrenmesi metotlarını bu beklentilere göre sınıflandırmak mümkündür;

- **Kümeleme:**Geçmişteki verilerin sınıfları/etiketleri verilmediği/bilinmediği durumlarda verilerin birbirlerine yakın benzerliklerinin yer aldığı kümelerin bulunmasıdır.

- **Sınıflandırma**: Geçmiş bilgileri hangi sınıftan olduğu biliniyorsa, yeni gelen verinin hangi sınıfa dâhil olacağının bulunmasıdır.
- **Regresyon - Eğri Uydurma**: Geçmiş bilgilere ait sınıflar yerine sürekli bilginin yer aldığı problemlerdir.

Eşler arası sistemlerde de makine öğrenmesi tekniklerinin uygulamaları mevcuttur. Kötü niyetli kullanıcıların karakteristiklerinin belirlenmesi, ağ trafiğinin eşit olarak dağıtılması, etkileşimlerin sonuçlarının önceden tahmini gibi problemler için makine öğrenmesine dayalı çalışmalar görülmektedir. Eşler arası sistemlerdeki güven yönetim probleminin kesin bir çözümü olmaması ve büyük miktarda veri ile birlikte geniş bir çözüm uzayını içermesi nedeniyle makine öğrenmesi tekniğine uygun bir problem oluşturmaktadır.

4.2 Genetik Programlama

Genetik programlama kavramı 1992 yılında John Koza [33] tarafından ortaya atılmıştır. Genetik algoritmaların yardımıyla çeşitli amaçlara hizmet eden ve evrimleşebilen programların geliştirilmesine genetik programlama adı verilmiştir.

Genetik programlamanın temeli Darwin'in evrim teorisine [34] dayanmaktadır. Bu da "güçlü olanın hayatta kalması"dır. Genetik programlama, birbirinden üretilen ve çözüme daha yakın programlar(bireyler) kümesinden oluşmaktadır. Her adımda bir öncekinden daha başarılı programların üretilmesi hedeflenmektedir. Zaman içerisinde bulunulan ortamın gereklerine göre en iyi olan program varlığını devam ettirmektedir.

Bilgisayar bilimlerindeki bazı problemler, karmaşıklık düzeyi yüksek ve çözümleri için geniş bir arama uzayına sahip problemlerdir. Bu denli geniş bir arama uzayı içerisinde, probleme uygun ve istenilen kıstasları

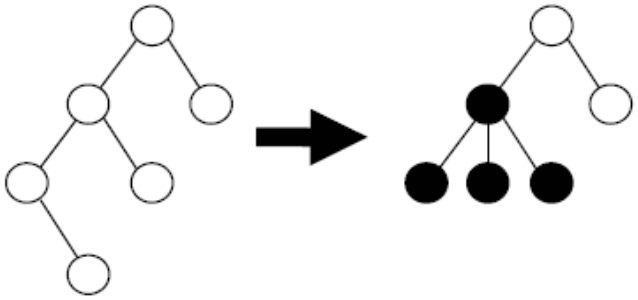
Şekil 10: Genetik Programlama Genel Mimarisi

sağlayan çözümü bulmak oldukça zor bir problem teşkil etmektedir. Genetik programlama, çözülmesi zor problemleri çözmek ya da en iyi çözüme yaklaşmak için bir yol sağlamaktadır[35]. Genetik programlama ile oluşturulan programların genel yapısını genetik operasyonlar, terminal ve fonksiyon kümeleri ile uygunluk fonksiyonları oluşturulmaktadır. Şekil 10, genetik programlamanın genel yapısını göstermektedir.

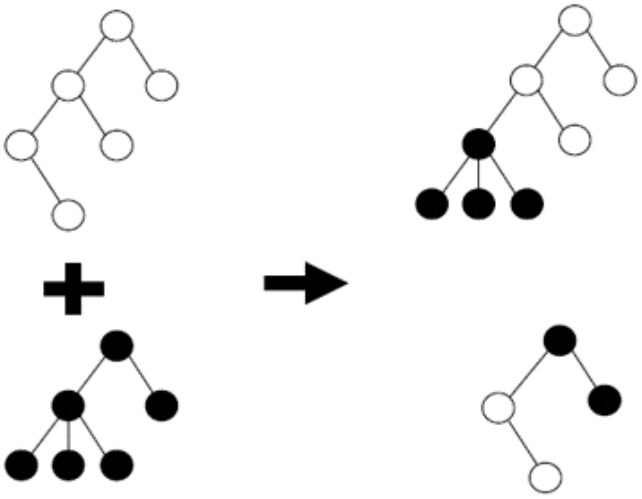
4.2.1 Genetik Operasyonlar

Genetik programlama bilgisayar programlarını barındıran ağaç yapıları üzerine çalışmakta ve bunlar üzerinde yaptığı işlemlere göre çözüme ulaşmayı hedeflemektedir. Ağaçların dalları üzerinde uygulanan genetik operasyonlar sonucunda bu hedeflere ulaşılmaktadır. Uygulanan genetik operasyonlar ise [36];

- **Çoğaltma(*Reproduction*):** Popülasyonda var olan bir programın yeni bir popülasyona aktarma işlemine çoğaltma denilmektedir.
- **Mutasyon(*Mutation*):** Ağaç yapısındaki bir düğümün yerine rastgele bir düğüm seçilerek gerçekleştirilen mutasyon işlemidir. Seçilen düğümün altında rastgele bir yapı büyütülür. Şekil 11 mutasyon işlemini göstermektedir.
- **Çaprazlama(*Crossover*):** Seçilen iki ağaç yapısının rastgele belirlenmiş iki düğümünün birbiri ile değiştirilmesi işlemine çaprazlama



Şekil 11: Mutasyon Operasyonu



Şekil 12: Çaprazlama Operasyonu

denilmektedir. Belirlenen düğümlerin alt ağaçları karşılıklı olarak takas edilmektedir. Şekil 12 çaprazlama işlemini göstermektedir.

4.2.2 Fonksiyon ve Terminal Kümesi

Programların genetik işlemlere tabi tutulması ve sonuçlara ulaşılabilmesi adına fonksiyon ve terminal kümesinin ifade edilmesi gerekmektedir. Bunun için fonksiyonlar ve terminal kümeleri kullanılmaktadır. Problemlerin çeşitlerine göre fonksiyon kümeleri ve terminaller farklılıklar göstermektedir. Önemli olan doğru çözüme götürebilecek olan kümelerin kullanılmasıdır.

Genelleştirilmiş problemlerin çözümü için fonksiyon kümesi toplama, çıkartma, çarpma ve bölme gibi aritmetik operatörlerden oluşabilir. Aynı

şekilde terminal kümesi de bağımsız değişkenler ve sayısal sabitleri içerebilir. Bu tip oluşturulmuş terminal ve fonksiyon kümeleri, geniş bir aralıktaki problemlerin çözümlerine uygulanabilmektedir [33].

Özelleştirilmiş problemlerin çözümü için ise alışılagelmiş fonksiyon ve terminal kümelerinin dışına çıkılabilmektedir. Probleme özel kabiliyetler ve değerlerin programların yapısına aktarılması ve çözüme ulaşılması gerekmektedir.

4.2.3 Uygunluk Fonksiyonu

Genetik programlamada programların iyi çözümlere ulaşabilmesi için yön tayini sağlayan fonksiyona uygunluk fonksiyonu (*fitness function*) denilmektedir. Uygunluk fonksiyonunun seçimi, probleme ve amaca göre farklılıklar göstermektedir. İyi seçilmiş bir uygunluk fonksiyonu, uygun çözümlerin bulunmasını sağlarken, kapsamlı düşünülmemiş bir uygunluk fonksiyonu modelin çözüm performansını doğrudan azaltacak bir etken oluşturmaktadır [33].

Uygunluk fonksiyonu oluşturulurken ulaşılmak istenen amaç, doğrudan uygulanabileceği gibi amaca eklenecek çeşitli terimler ile de çözümün iyi performanslı sonuçlara yaklaştırılması sağlanabilmektedir [37]. Uygunluk fonksiyonunun belirlenmesinden önce ulaşılmak istenen amaç, tam olarak nitelendirilmeli ve amaca göre çözümü bulmaya yardımcı olacak uygunluk fonksiyonu oluşturulmalıdır.

4.3 Makine Öğrenmesinin Güven Yönetiminde Kullanılması

Liu [38] tarafından geliştirilen yapıda dağıtık çalışan sistemler için makine öğrenmesi tekniği kullanılarak genel bir model önerilmektedir. Önerilen

model kullanıcıların diğer kullanıcılar ile yaptığı geçmiş işlemlere bakarak yeni yapacağı işlemlerin sonucunu öngörmeye çalışmaktadır. Bunun için çıkardığı öznitelik (*features*) kümesini kullanarak makine öğrenmesi teknikleriyle işlemleri başarılı ya da başarısız olma ihtimaline göre sınıflandırmaktadır.

Çalışmada kurulan genel yapının birçok makine öğrenmesi tekniği üzerinde uygulanabilir olduğu belirtilmiştir. Örnek olarak iki makine öğrenmesi tekniği üzerinde uygulanmıştır. Bunlar; boyut küçültme ve sınıflandırmada etkili olan Doğru Bileşenler Analizi (*Linear Discriminant Analysis*) [39, 40] ve ağaç yapısını kullanarak sınıflandırma ve tahmin üzerine çalışan Karar Ağacı (*Decision tree*) [41] teknikleridir.

Kullanıcıların geçmiş bilgilerinden gelecekte yapacakları davranışları tahmin etmek üzere var olan güven yönetim sistemlerinin aksine, önerilen model sadece yerel güven bilgisini kullanmaktadır. Bunun dağıtık çalışan sistemlerde ağ trafiğini düşürdüğünü ve üçüncü parti yazılımların güven sorununu ortadan kaldırdığını savunmaktadırlar.

Yerel bilgiye sahip olmayan kullanıcılar için önerdikleri yerel bilgi paylaşım ağı yapısı sayesinde yerel bilgilerin kullanıcılar arasında güvenli bir şekilde paylaşıldığını belirtmektedirler. Bu ağ yapısı sayesinde tavsiye ve geri besleme yöntemine dayalı güven yönetim sistemlerinden farklı olarak bazı avantajları olduğu belirtilmektedir. Bunlar;

- Sahte bilgilerin paylaşım ihtimalinin azaltılması
- Gizlilik sızıntısının önlenmesi
- Fazlaca yapılacak hesaplamaların ve iletişimin getireceği maliyetin azaltılması

Çalışmada genel olarak sadece yerel güven bilgilerini kullandıklarını belirtseler de yerel bilgi paylaşım yapısı diğer kullanıcıların bilgilerine de

ihtiyaç olduğunu göstermektedir. Yerel bilgilerin bu şekilde paylaşılması da zaten hesaplama ve iletişim maliyeti getirecektir. Kötü niyetli bir kullanıcı bu paylaşım yapısını da aldatarak sahte yerel bilgiler verebilir.

Beverly [42] tarafından önerilen modelde makine öğrenmesi tekniğini kullanarak hem iletişim maliyetini düşürmeye hem de ileride olması beklenen sorgulara cevap verebilecek komşu kullanıcıları tespit etmeye çalışılmaktadır. En büyük başarı ve en az sorgu ile sonuca ulaşmayı hedeflemektedir. Temelde Destek Vektör Makinesi'ni [43, 44] (*Support Vector Machines*) kullanmaktadır. Makine öğrenmesinde kullanılacak öznitelikleri (*features*) seçerken hesaplama karmaşıklığı ve zorluğu değil iletişim maliyetinin düşürülmesi hedeflenmiştir. Bu özelliklere göre komşu seçimi yapılmaktadır.

Çalışmada Gnutella ağından toplanmış büyük çapta veriler kullanılmakta ve bu veriler anlamlandırılarak modelin başarımı sağlanmaktadır. Model iletişim maliyetlerini düşürürken amaca yönelik komşuların bulunmasını sağlamaktadır.

Song [45] tarafından önerilen çalışmada dağıtık bir itibar yönetim sistemi ve global bir itibar modeli ortaya atılmaktadır. Bu model, temelinde efendi-köle (*master-slave*) ilişkisini barındırmaktadır. Global itibar modeli, Yapay Sinir Ağları (*Artificial Neural Networks*) [46] tekniğinin dağıtık itibar sistemlerine uygulanması temeline dayandırılmaktadır.

Modelin yapısında bir tane global efendi ajan (*Global master agent*) ve onun kontrolünde olan yerel köle ajanlar (local slave agent) bulunmaktadır. Yerel köle ajanlar belirli bir kullanıcı grubunu gözlemlemekte ve modeli bu gruplar üzerinde uygulamaktadırlar. İtibarın yerel ajanlar arasında toplanması için HISTOS [47] modelini kullanmaktadırlar. Global efendi ajan tarafında ise üç işlem koşmaktadır. Bunlardan ilki, global itibar modelidir. Global itibar modeli sadece bir kullanıcı hakkında birden fazla yerel itibar sorgusu geldiğinde çalışmakta ve sonuç üretmektedir. İkincisi

merkezi itibar modelidir. Bu model kullanıcıların birden fazla yerel itibar değerlendirmelerini toplamakta ve merkezi değerlendirme algoritmasını çalıştırmaktadır. Toplanan sonuçlar ise kullanıcının yapay sinir ağlarındaki global itibarını hesaplamakta kullanılmaktadır. Son olarak üçüncüsü ise izleme sürecidir. İzleme süreci, genel olarak grupların ve global itibarın performansını gözlemlemektedir.

Önerilen modelde, ağ kullanımı sırasında Yük Dağılımı [48] (*Load Balancing*) teknikleri kullanarak ağ trafiği azaltılmaya çalışılmaktadır. Ayrıca itibar değerlendirmesini hesaplamak için ayrılan zamanı, ajanlar arasında yaptığı iş paylaşımı ile azaltmayı hedeflemektedir.

Song [49] tarafından ortaya atılan model tavsiyeler ve güven değerlendirme modellerinin sonuçlarından güven değerini bulmaya çalışan heterojen bir yapı sunmaktadır. Mimarisi tavsiye edenler (*recommenders*) üzerine kurulu olan model, tavsiye edenleri nitelikli ve niteliksiz olarak ayırmaktadır. Sistem nitelikli olarak adlandırdığı tavsiye edenler üzerine yoğunlaşarak güveni sağlamaya çalışmaktadır. Bir kullanıcının nitelikli ya da niteliksiz olarak sınıflandırılması diğer kullanıcıların güven düşüncesine göre yapılmaktadır. Güven düşünceleri yapay sinir ağını oluşturmak için eğitim verisi olarak kullanılmaktadır. Seçimler öncelikle nitelikli kullanıcılar arasında yapılan sıralamaya göre gerçekleştirilmektedir. Eğer nitelikli *recommender* bulunamazsa, kullanıcıların güven değerlerine göre bir sıralama yapılmaktadır. İşlemleri kendi içinde sınıflayan ve parçalara ayıran model genel olarak hızlı bir çalışma ve hafifletilmiş bir ağ trafiği yükü getirmektedir. Ayrıca farklı yapılara uygulanabilirliği de modelin bir avantajı olarak belirtilmektedir.

Selveraj [50] tarafından ortaya atılan çalışmada kullanıcı profillerini kural tabanlı bir yapıya oturtup Genetik Algoritmalar yardımı ile modellemeyi içermektedir. Modelde her bir kullanıcı kendi güven yönetimini yapmaktadır. Bunun için kullanıcıların normal davranışları izlenmekte ve aykırılık tespiti

(*anomaly detection*) yöntemi ile normal olmayan davranış profilleri analiz edilmeye çalışılmaktadır.

Çalışma mantığı olarak iki aşamada işlemler gerçekleşmektedir. Bunlardan ilki öğrenme aşamasıdır. Bu aşamada normal kullanıcı profilleri öğrenilmeye çalışılmaktadır. Bunun için Denning [51]'in kural tabanlı modeli kullanılmaktadır. Çalışmada önerilen model, kullanıcı davranışlarını kural tabanlı tanımlamakta ve buna göre kararlar vermektedir. Öğrenme aşamasında sadece normal davranışlı kullanıcıların olduğu varsayılmıştır. Bu sayede normal davranışlı kullanıcı modeli tam olarak ortaya konulmaya çalışılmıştır. İkinci aşamada ise güven değerlendirme adımına geçilmektedir. Bu aşamada aykırılık tespiti yöntemi ile genetik algoritmaları da kullanarak şüpheli davranışlar bulunmaya çalışılmaktadır. Her bir davranış, belirlenen uygun fonksiyonundaki sınır değerine göre normal ya da anormal olarak nitelendirilmektedir. Bu sayede anormal olan davranışların sistemden uzaklaştırılması hedeflenmektedir.

Kullanıcı profilleri tanımlaması iki tür parametre kümesi ile ifade edilmiştir. Bunlar; bağlantı parametreleri ve oturum parametreleridir. Bu parametreler ikili sayılara dönüştürülerek genetik algoritmaya verilmiş ve kullanıcı davranış profillerinin modellenmesi sağlanmıştır. Uygunluk fonksiyonu sayesinde 0 ile 1 arasında modellenmiş davranışların 0,5 ve üstü çıkması durumunda normal olarak nitelendirilmiştir. Sistemde tüm kullanıcıların bilgilerini tutan ve tekil kimlik numarası veren bir merkezi otorite olduğu varsayılmıştır. Bu sayede kullanıcıların kimlik doğrulama işlemlerinin güvence altına alındığı belirtilmiştir. Tüm kullanıcıların yönetimi bu otorite tarafından sağlanmaktadır.

Çalışmada kullanıcılar kendi güven yönetimlerini kendileri yapmaktadır. Kullanıcılar kendi aralarında güven verilerini paylaşmamaktadırlar. Bu da sistemin güven mekanizmasının zayıf kalmasına sebebiyet vermektedir. Ayrıca merkezi bir otoritenin varsayılması ve bir takım güvenlik sorunlarının

bu şekilde çözülmesi modelin gerçekçiliğini azaltmaktadır. Eşler arası sistemlerin doğası gereği, bağımsız ve dağıtık çalışma mantığına merkezi bir otorite ters düşmektedir.

Güven yönetiminde genetik algoritmaların az sayıda uygulaması olmasına rağmen, son yıllarda evrimsel hesaplama yöntemleri bilgisayar ve ağ güvenliğinde başarı ile kullanılmaya başlanmıştır. Özellikle, saldırı tespitinde, bizim problemimize benzer şekilde, etkin olarak kullanılmıştır. Genetik programlama veya genetik algoritmalar saldırı tespitinde kullanılan yöntemler olmuştur. Saldırı tespitinde genetik programlama ile yapılan ilk çalışma Crosbie ve Spafford [52] tarafından gerçekleştirilmiştir. Bu çalışma sonrasında da bu alanda başarılı bir çok çalışma daha gerçekleştirilmiştir. Abraham ve Grosan [53] saldırı tespitinde genetik programlamayı diğer bazı makine öğrenmesi yöntemleri (destek vektör makineleri ve karar ağaçları) ile karşılaştırmış ve genetik programlamanın diğer yöntemlerden daha başarılı olduğunu göstermişlerdir. Dilbilimsel evrim (*grammatical evolution*) teknikleri de kablolu [54] ve kablosuz tasarsız ağların [55] saldırı tespitinde son yıllarda başarı ile uygulanmıştır. Sen ve Clark [56], tasarsız ağlar gibi kaynak kısıtı olan ağlardaki saldırıların tespitinde, hem tespit oranı yüksek, hem de az enerji harcayan tespit programlarının evrimsel hesaplama ile üretilebildiğini göstermiştir [56]. Evrimsel hesaplamanın, saldırı tespit sensörlerin yerleşimi [57], saldırı tespit sistemlerindeki özniteliklerin belirlenmesi/azaltılması [58] gibi uygulamaları da bulunmaktadır. Tüm bu uygulamaların başarımı, evrimsel hesaplamanın güven yönetimindeki başarısının araştırılması için bir motivasyon sağlamıştır.

5 MODEL VE YÖNTEM

Bu tez kapsamında yapılan çalışmada eşler arası sistemlerdeki güven yönetimi probleminin genetik programlama ile sağlanması hedeflenmektedir. Bu kapsamda oluşturulan modüller ve gerçekleştirilen işlemler anlatılacaktır.

5.1 Genel Yapı

Eşler arası sistemlerde güven yönetimini sağlamak çözülmesi zor bir problemdir. Daha önceki yaklaşımlarda bu problemin çözümü için istatiksel ve matematiksel çözümlere başvurulmuştur. Çözüm uzayı geniş ve zor olan bu problem için sabit çözümler yeterli gelmemektedir. Çözülmesi zor probleme karşı kendini geliştirebilen ve iyileştirebilen bir model geliştirilmesi gerekmektedir. Bu zorluk göz önüne alınarak yapılan bu tez çalışması kapsamında, eşler arası sistemlerdeki güven yönetiminin genetik programlama yardımı ile kendini geliştirebilen ve saldırılara karşı şekillendirilebilen bir model önerilmiştir. Model, saldırılara karşı kendini eğitebilmekte ve yeni gelecek saldırıları geçmiş eğitimine dayanarak engelleyebilmektedir. Genetik programlamanın probleme uygun olarak belirlenmesi ve seçilmesindeki sebep birçok makine öğrenmesi tekniklerinden ya da insanların kendi geliştirdikleri programlardan daha başarılı sonuçlar üretmesi olarak açıklanabilir [59].

Modelin oluşturulması için kurulan genel yapı, iki ayrı modülden oluşmaktadır. Bunların ilki simülasyon modülü, ikincisi ise genetik programlama modülüdür. Simülasyon modülü, eşler arası bir sistemin benzetimini sağlamaktadır. Bu modülde iyi niyetli ve kötü niyetli kullanıcıların belirli bir zaman için davranışları modellenmektedir. Genetik programlama modülünde ise genetik programlamanın operasyonlarını gerçekleştirilmekte ve en iyi sonuca ulaşılmaya çalışılmaktadır. Genetik programlama modülü

sayesinde güven değerinin hesaplanacağı çözüm bulunmakta ve bu çözüm simülasyon modülündeki kullanıcı etkileşimlerinde kullanılmaktadır. Yapı, iki modülün entegre olması ile oluşmaktadır. Modüller, ileride detaylı olarak anlatılacaktır.

Tez kapsamında önerilen model, genetik programlamanın öğrenme ve test etme evreleri ile şekillenmektedir. Çeşitli saldırılara karşı önce eğitilen model daha sonra yeni bir ortamda test edilmektedir. Testlerdeki başarısı modelin genel başarımı olarak nitelendirilmektedir. Modelin şekillenmesini sağlayan bu evrelerden ilerde detaylı olarak bahsedilecektir.

Çalışma temel olarak sırası ile aşağıdaki adımlar üzerinden gerçekleştirilmiştir;

- Simülasyon ve genetik programlama modüllerinin oluşturulması,
- Genetik programların yapısını oluşturacak özniteliklerin belirlenmesi,
    - Terminal ve fonksiyon kümesinin belirlenmesi,
    - Uygunluk fonksiyonunun belirlenmesi,
- Simülasyon ve genetik programlama parametrelerinin belirlenmesi,
- Modeli eğitme,
- Test ve sonuçları değerlendirme.

5.2 Simülasyon Modülü

Güven yönetimini sağlayabilmek adına öncelikle gerçeğe yakın eşler arası bir sistemin modellenmesi gereği doğmaktadır. Bu gereği karşılamak için ilk olarak eşler arası sistemi benzeten bir simülasyon, [60] çalışması temel alınarak oluşturulmuştur. Temel alınan çalışmanın [60] yaklaşık %30 oranında değiştirilmesi ve modelin yapısına uyarlanması ile birlikte

simülasyon modülü geliştirilmiştir. Simülasyon, eşler arası sistemlerin amacı ile aynı amaçta çalışarak kullanıcıların ihtiyaçlarını karşılamayı hedeflemektedir. Modül, JAVA programlama dili kullanılarak geliştirilmiştir.

Simülasyon, saf eşler arası sistemlerde olduğu gibi kullanıcılardan oluşmaktadır. Eşler arası bir sistemin gereği olarak kullanıcılar birbirleri aralarında dosya alıp vermektedirler. Kullanıcılar genel olarak iki sınıfta modellenmiştir. Bunlardan ilki sistemde bulunma amacı sadece aradığı dosyalara ulaşmak olan iyi niyetli kullanıcılardır. Bu kullanıcılar her zaman sağlıklı dosyalar paylaşırlar. İkinci kullanıcı grubu ise kötü niyetli kullanıcılardır. Kötü niyetli kullanıcılar davranış türlerine göre çeşitlilikler arz etmektedirler. Temel amaçları sisteme olabildiğince zarar vermek ve varlıklarını sürdürebilmektir.

Tüm kullanıcılar simülasyonun başlangıcında birbirlerine yabancı olarak sisteme dâhil olmaktadırlar. Birbirleri arasında gerçekleşen dosya indirme işlemleri ile tanınmaya başlamaktadırlar. İki kullanıcı arasındaki dosya indirme işlemi “etkileşim” olarak adlandırılmaktadır. Kullanıcılar sistemde var oldukları sürece güncel olan belirli sayıdaki etkileşim geçmişlerini tutmaktadırlar. Bir kullanıcı başka bir kullanıcıdan dosya indirdiğinde dosyanın içeriği bozuk ya da virüslü olup olmamasına göre etkileşimi başarılı ya da başarısız olarak nitelendirebilmektedir. Eğer etkileşimleri başarılı olarak nitelendirilirse dosya indirdikleri kullanıcı ile komşu(dost) olmaktadırlar. Eğer virüslü ya da içeriği bozuk dosya alırlarsa etkileşim başarısız olarak nitelendirilmektedir. Kullanıcılar, başarısız etkileşimler sonucunda etkileşime girdikleri kullanıcıyı kötü niyetli olarak işaretlemekte ve bir daha asla o kullanıcı ile etkileşime girmemektedirler. Başarısız etkileşimler, sadece virüs veya bozuk içerikli dosyaların paylaşılması sonucu gerçekleşen etkileşimlerdir. Dosya paylaşımı sırasında oturumun herhangi bir sebepten iptal olması başarısız etkileşim olarak nitelendirilmemektedir. Bu tip iptal durumları sadece kullanıcı hakkında olumsuz değerlendirmeye sebebiyet vermektedir.

Kullanıcılar tanımadıkları bir kullanıcı ile etkileşime girecekleri zaman komşularından o kullanıcı hakkındaki görüşlerini almaktadırlar. Bu görüşlere "tavsiye" adı verilmektedir. İyi niyetli kullanıcılar her zaman dürüst tavsiyeler verirken, kötü niyetli kullanıcılar dürüst olmayan tavsiyeler verebilmektedirler. Bu kötü tavsiyeler, sistemde "yanıltıcı tavsiye" olarak nitelendirilmektedir. Özetlemek gerekirse iyi niyetli kullanıcılar devamlı virüssüz dosya ve dürüst tavsiyeler verirken, kötü niyetli kullanıcılar karakteristiklerine göre virüslü veya içeriği bozuk dosya ya da yanıltıcı tavsiyeler verebilmektedirler ve bunlar saldırı olarak nitelendirilmektedir.

Simülasyonun gerçek eşler arası sisteme benzetilmesi için kullanıcı davranışları gerçeğe yakın modellenmeye çalışılmıştır. Kullanıcılar sistemde rastgele olarak çevirim içi ve çevirim dışı olabilmektedirler. Dosya indirme işlemleri oturum olarak nitelendirilmektedir. Oturumlar sırasında eğer kullanıcılardan biri çevirim dışı olursa belirli bir süre indirme işlemi bekletilmektedir. Eğer oturum zaman aşımına uğrarsa dosya indirme işlemi iptal edilmekte ve dosyayı paylaşan kullanıcı için negatif izlenim bırakmaktadır.

Simülasyon modülünün davranışını gerçeğe yaklaştırmak adına birçok farklı parametre ele alınmıştır. Bu parametreler birçok deneysel çalışmaların sonuçlarına göre belirlenmiştir [61, 62, 63, 64]. Çizelge 1 kullanılan parametreleri göstermektedir. Parametrelerin sahip olduğu oranlar, o parametrenin verilen değerleri için sistemde yüzde kaç oranında var olacağını göstermektedir. Bu parametrelerden dosya boyutu sistemde var olan dosyaların boyutlarının hangi oranlarda olacağını göstermektedir. Başlangıç yükleyici dosya dağılımı parametresi ise bir yükleyicinin simülasyonun başlangıcında kaç tane dosya paylaşabileceğinin oranlarını göstermektedir. Paylaşılan dosya dağılımı parametresi bir dosyanın kaç kullanıcı tarafından paylaşılabileceğinin oranlarını gösterirken, bantgenişliği dağılımı ise kullanıcıların dosya indirme ve yüklemede sahip olabilecekleri bantgenişliklerinin oranlarını göstermektedir. Son olarak çevirim içi

Çizelge 1: Simülasyonda Kullanıcı ve Kaynak Girdilerini Oluşturan Temel Parametreler

(a) Dosya Boyutu Dağılımı

Dosya Boyutu (kb)	Oran
100 - 1000	0.10
1001 - 10000	0.75
10001 - 100000	0.10
100001 - 1000000	0.05

(b) Başlangıç Yükleyici Dosya Dağılımı

# Yükleyici	Oran
1 - 5	0.60
6 - 10	0.20
11 - 20	0.15
21 - 40	0.05

(c) Paylaşılan Dosya Dağılımı

# Paylaşılan Dosya	Oran
0 - 0	0.20
1 - 20	0.25
21 - 100	0.40
101 - 200	0.10
201 - 400	0.05

(d) Bantgenişliği Dağılımı

İndirme-Yükleme Bant genişliği (kbps)	Oran
128 - 64	0.10
512 - 128	0.10
1024 - 256	0.40
3036 - 768	0.20
10240 - 5120	0.15
102400 - 10240	0.05

(e) Çevirimiçi Periyot Dağılımı

Çevirim içi (Dk)	Oran
61 - 120	0.30
121 - 240	0.50
241 - 360	0.10
361 - 600	0.05
601 - 720	0.05

(f) İndirme Oturumu İçin Bekleme Periyodu

Dosya Boyutu (kb)	Mak. Bekleme Periyot
100 - 1000	1
1000 - 10000	3
10000 - 100000	10
100000 - 1000000	100

periyot dağılımı paremetresi kullanıcıların çevirim içi sürelerinin hangi oranlarda olacağını gösterirken indirme oturumu için bekleme periyodu paylaşılan boyutlardaki dosyalar göre indirme işlemi için maksimum bekleme periyotlarını göstermektedir.

Kullanıcıların birbirleri arasında yaptığı etkileşimlerin çeşitli özellikleri bulunmaktadır. Bunlardan ilki “memnuniyet” değeridir. Bu değer kullanıcının etkileşimden ne kadar memnun kaldığına göre değer almaktadır. Memnuniyet değeri, ortalama bant genişliği (*AveBw*), kararlaştırılmış bant genişliği (*AgrBw*) ve kullanıcıların işlem sırasındaki çevirim içi (*OnP*) ve çevirim dışı (*OffP*) zamanlarını dikkate almaktadır. Memnuniyet [60] çalışmasında olduğu gibi aşağıdaki formüle göre hesaplanmaktadır:

$$\textbf{Memnuniyet} = \begin{cases} (\frac{AveBw}{AgrBw} + \frac{OnP}{OnP+OffP})/2 & if\ AveBw < AgrBw, \\ (1 + \frac{OnP}{OnP+OffP})/2 & otherwise \end{cases} \quad (5.1)$$

Etkileşimin diğer bir özelliği ise etkileşim sonucunda alınan "Ağırlık" değeridir. Bu değer yapılan dosya indirme işlemine göre, dosya boyutu (*size*) ve kullanıcı sayısı (*uploaders, UploaderMax*) ile ilişkilidir. Ağırlık değeri [60] çalışmasında olduğu gibi aşağıdaki formüle göre hesaplanmaktadır:

$$\textbf{Ağırlık} = \begin{cases} (\frac{size}{100MB} + \frac{\#Uploaders}{Uploader_{max}})/2 & if\ size < 100MB, \\ (1 + \frac{\#Uploaders}{Uploader_{max}})/2 & otherwise \end{cases} \quad (5.2)$$

Simülasyonda her bir kullanıcı, ileride anlatılacak olan genetik programlama yardımı ile bulunmuş denklem sayesinde hesaplanan güven metriği puanı ile değerlendirilmektedir. Dosya indirme işlemi öncesi geçmiş etkileşimler ve alınan tavsiyeler ışığında güven metriği puanı hesaplanmaktadır. Hesaplama sonrasında, öncelik güvenilen komşulara verilecek şekilde, güven metriği puanına göre kullanıcıların sıralaması yapılmaktadır. Bu sıralamaya göre ise etkileşim başlatılmaktadır. Fakat sıralamaya girebilmek için en önemli kural, dosya arayan kullanıcı ile daha önceden virüslü veya içeriği bozuk bir dosya paylaşımı sebebiyle oluşan ve başarısız olarak nitelendirilen bir etkileşime girmemiş olmaktır. Eğer başarısız bir etkileşime girilmemişse güven metriği puanına bakılmaktadır. Eğer sıralamadaki kullanıcıların güven metriği puanları da eşitse, dosya yükleme hızlarına bakılmaktadır. Bu da eşitse rastgele bir kullanıcı seçilmektedir.

Simülasyon modülü, hem bir güven modeline dayalı olarak hem de güven modeli olmadan çalışabilmektedir. Bu sayede, sümülasyonun güven modeli olmayan bir ortamda çalıştırılmasıyla alınan sonuçlar ile güven modeli

uygulandığında alınan sonuçlarının karşılaştırılabilmesi sağlanmaktadır. Bu da ileride anlatılacak genetik programlama modülünde bulunan çözümlerin iyileştirilmesinde kullanılmaktadır.

Algoritma 1 Simülasyon Modülünün Genel Çalışma Adımları

```
kullanıcıları yarat ve ortamı oluştur
rastgele olarak kullanıcıların çevirim içi ve çevirim dışı durumlarını ayarla
while şimdiki periyot <= maksimum periyot do
  for all tüm kullanıcılar do
    if kullanıcı çevirim içi then
      for all kullanıcının devam eden tüm oturumları do
        if yükleyici sistemde yok then
          oturumu iptal et
          yükleyicinin etkileşim değerlerini olumsuz güncelle
        else
          indirilen dosya boyutunu hesapla
          if dosya tamamlandı then
            oturumu sonlandır
            etkileşimi başarılı ya da başarısız olarak değerlendir
            yükleyicinin etkileşim değerlerini güncelle
          else if oturum süresi doldu then
            indirme işlemini iptal et
            yükleyicinin etkileşim değerlerini olumsuz güncelle
          end if
        end if
      end for
      if güven modeli var ve güven değeri güncelleme periyodu then
        kullanıcının komşularının güven değerlerini güncelle
      end if
    end if
  end for
  if kimlik değiştiren saldırganlar var ve kimlik değiştirme periyodu then
    kimlik değiştiren saldırganların kimlerini değiştir
  end if
  for all tüm kullanıcılar do
    if kullanıcı çevirim içi then
      rastgele bir dosya için indirme işlemi başlat
    end if
  end for
end while
```

Yukarıda anlatılanların ışığında simülasyon modülünün çalışma adımları Algoritma 1 ile gösterilmiştir. Başlangıçta tüm kullanıcılar hiçbir komşuluk ilişkisi olmadan sisteme dâhil olmaktadırlar. Simülasyonun başlaması ile

birlikte kullanıcılar, her bir periyotta çevirim içi durumlarına göre işlem yapmaktadırlar. Çevirim içi olan bir kullanıcı sırası geldiğinde var olan oturumlarının durumlarını güncellemektedir. Eğer dosya indirdiği kullanıcı ile komşuluk ilişkisi bir şekilde bozuldu ise indirme işlemini iptal etmekte ve yükleyicinin güven değerini kötü olarak etkilemektedir. Eğer indirme işlemi tamamlanır ve etkileşim başarılı olursa memnuniyet ve ağırlık değerleri hesaplanarak yükleyici hakkındaki etkileşim değerleri olumlu olarak güncellenmektedir. Fakat içeriği bozuk ya da virüslü bir dosya indirilmesi halinde yükleyici kötü niyetli bir kullanıcı olarak belirlenip o yükleyici ile tekrardan asla etkileşime girilmemektedir. Etkileşimin değerlendirilmesi ve başarılı olması halinde indirilen dosya rastgele yapılan bir şeçim sonrasında kullanıcı tarafından paylaşılabilmektedir. Kullanıcının tüm oturumları o periyot için değerlendirildikten sonra, eğer simülasyonda güven modeli uygulanıyor ve güven değeri güncelleme periyodu geldi ise kullanıcının tüm komşularının güven değeri hesaplanarak güncellenmektedir. Hesaplama esnasında komşusu için diğer komşularından tavsiyeler almakta ve o komşu ile gerçekleştirdiği kendi etkileşim bilgilerini de hesaba katarak ileride anlatılacak genetik programlama modülü ile bulduğu denklem sayesinde komşusunun güven değerini güncellemektedir. Tüm kullanıcıların o periyot için oturumlarının ele alınmasının ardından eğer sistemde kimlik değiştiren saldırganlar var ve kimlik değiştirme periyodu geldi ise saldırganlar kimliklerini değiştirmektedirler. Son olarak çevirim içi olan tüm kullanıcılar için rastgele bir dosya seçilip indirme oturumu başlatılmaktadır. Kullanıcılar rastgele seçtikleri bir dosyayı indirecekleri zaman dosyayı indirecekleri kullanıcı olarak önceliklerini komşularına vermektedirler. Komşuları arasında güven değerine göre yapacakları bir sıralama ile en güvendikleri komşularından dosya indirme oturumu başlatmaktadırlar. Eğer indirilecek dosya komşularda bulunamaz ise yabancı kullanıcılar ile oturum başlatılmaktadır. Kullanıcı, indirilecek dosyaya sahip yabancı kullanıcılar için kendi komşularından tavsiyeler almakta ve güven değerini hesaplayarak en güvenli gördüğü yabancı

kullanıcı ile oturum başlatmaktadır. Simülasyon bundan sonra bir sonraki periyoda geçmekte ve işlemler tüm periyotlar sonlanana kadar aynı şekilde devam etmektedir.

5.3 Genetik Programlama Modülü

Eşler arası sistemlerde güven yönetimini sağlamak ve kötü niyetli kullanıcıları sistemden uzaklaştırmak adına sabit formüller ya da istatiksel yöntemlere dayalı yaklaşımlar literatürde çalışılmıştır. Bu çalışmada, doğası gereği çeşitlilik gösteren ve sürekli değişerek gelişen bu problemi çözmek için genetik programlama yaklaşımı önerilmiştir. Bu yüzden, simülasyon modülünü kullanan ve sarmalayan genetik programlama modülü geliştirilmiştir. Genetik programlama modülünün güven değerini hesaplamak adına bulduğu çözümler, simülasyon modülünde kullanılmakta ve saldırganların yaptığı saldırılar azaltılmaya çalışılmaktadır.

Genetik programlama modülü, temelde ECJ 21 [65] aracını kullanmaktadır. ECJ 21 aracı sayesinde genetik programlamanın tüm adımları gerçekleştirilmektedir. Araç sayesinde popülasyonlar yaratılmakta, genetik operasyonlar işletilmekte ve uygunluk fonksiyonuna göre sonuçlar iyileştirilebilmektedir. Terminal ve fonksiyon kümesinin, kullanılacak özniteliklerin ve uygunluk fonksiyonunun tanımlanmasının ardından ECJ 21 aracı tüm genetik programlama adımlarını verilen parametrelere göre kendisi gerçekleştirmektedir. ECJ'nin kullanılması sayesinde bireylerin mutasyona uğrama yüzdesi, çaprazlanacak birey sayısı, popülasyondaki birey sayısı, işleme tabi tutulacak jenerasyon sayısı vs. gibi birçok parametre değiştirilebilir ve probleme göre uygulanabilir kılınmıştır.

Modül, ileride bahsedilecek öznitelikleri kullanarak belirtilen terminal ve fonksiyon kümelerinin yardımıyla uygunluk fonksiyonunu sağlayan en iyi çözümü bulmaya çalışmaktadır. Bu çözüm daha önce bahsedilen

simülasyon modülünde kullanılacak güven metriği değeri hesaplama denklemini içermektedir. Genetik programlama modülünün her bir bireyinin oluşturduğu denklemler, simülasyon modülü üzerinde çalıştırılmaktadır. Simülasyon modülündeki saldırıları azaltma başarısına göre bireyler genetik operasyonlara tabi tutulmakta ve sonuçlar iyileştirilmeye çalışılmaktadır. Genetik programlama modülünün genel çalışma algoritması aşağıda verilmiştir.

Algoritma 2 Genetik Programlama Modülü Çalışma Adımları

```
rastgele bir popülasyon oluşturulur
while şimdiki jenerasyon <= maksimum jenerasyon do
  for all şimdiki jenerasyondaki bireyler do
    simülasyonu çalıştır
    uygunluk fonksiyonuna göre değerlendir
  end for
  genetik operasyonları uygula
  yeni bir popülasyon yarat
end while
```

Genetik programlama modülü, öncelikle simülasyon ortamını güven modeli olmadan çalıştırmakta ve oluşan saldırı sayısını ileride anlatılacak uygunluk fonksiyonunda kullanmak için saklamaktadır. Bu işlemin ardından parametre olarak verilen birey sayısı kadar birey içeren ilk popülasyon, genetik programlama modülü tarafından oluşturulmaktadır. Güven değerini hesaplayacak olan çözümleri içeren bu bireylerin her biri için simülasyon modülü baştan sona çalıştırılmakta ve simülasyonun güven değeri hesaplama işlemi genetik programlama modülünün o anki bireyine göre yapılmaktadır. Simülasyonun çalışmasının ardından oluşan saldırı sayılarına göre ileride anlatılacak uygunluk fonksiyonu işletilmektedir. Bu sayede, o anki popülasyondaki bireylerin verdiği uygun sonuçlara göre genetik operasyonlara tabi tutulmakta ve bir sonraki jenerasyon için çözüme daha uygun bireylerin oluşması sağlanmaktadır. Tüm jenerasyonlar için bu işlemleri gerçekleştiren genetik programlama modülü, en sonunda en başarılı bireyi(çözümü) göstermektedir.

Genetik programlamada öznitelikler kullanılarak, fonksiyonlar ve

terminaller GP ağaçlarını oluştururlar. Her bir GP ağacı popülasyondaki birer bireyi temsil etmektedir. Her bir nesilde oluşturulan bireyin başarısı problemi çözme yeteneğine ve uygunluk fonksiyonuna bağlıdır. Genetik programlama modülünde kullanılan bu anahtar yapılar aşağıda detaylandırılmaktadır.

5.3.1 Öznitelik Kümesi

Genetik programlamada doğru öznitelik kümesini oluşturmak oldukça zor bir problem olup hem genetik programlama hem de diğer makine öğrenmesi teknikleri ile alınacak sonuçlardaki başarının anahtarı niteliğini taşımaktadır [66]. Tez çalışması kapsamında modüllerin oluşturulmasının ardından ise öznitelik kümesinin oluşturulması adımı gerçekleştirilmiştir. Tez kapsamında yapılan çalışmalar ile oluşturulan öznitelik kümesi, iki ayrı başlıkta ele alınmıştır. Bunlar, geçmiş etkileşim tabanlı öznitelikler ve komşulardan alınan tavsiye tabanlı özniteliklerdir.

Etkileşim tabanlı öznitelikler kümesi bir kullanıcının diğer kullanıcılar ile olan geçmiş deneyimleri üzerine oluşturulmuştur. Bu deneyimler geçmişte iki kullanıcı arasındaki doğrudan gerçekleşen etkileşimler sonucu meydana gelmektedir. Eşler arası sistemlerin yapısına göre bu deneyimler dosya paylaşımı ya da işlemci paylaşımı gibi olaylar sonucu sağlanabilmektedir. Tez kapsamında ise etkileşim kümesi, dosya paylaşımındaki etkileşimin başarılı olup olmamasından oluşmaktadır. Bu öznitelikler, sistemdeki her etkileşime girmiş kullanıcı çiftleri arasında ayrı ayrı tutulmaktadır. Bu çalışmada, iki kullanıcı arasında gerçekleşen etkileşimlere göre tutulan öznitelikler ise şöyledir; etkileşim sayısı, başarılı etkileşim sayısı, indirilen ortalama dosya boyutu, etkileşimler arasında geçen ortalama zaman, her bir etkileşim sonucu ortaya çıkan ağırlık ortalaması ve her bir etkileşim sonrası ortaya çıkan memnuniyet ortalamasıdır. Çizelge 2 etkileşim tabanlı öznitelikler kümesini göstermektedir.

Çizelge 2: Etkileşim Tabanlı Öznitelikler

Öznitelik	Sembol
Etkileşim sayısı	f1
Başarılı etkileşim sayısı	f2
İndirilien ortalama dosya boyutu	f3
Etkileşimler arasında geçen ortalama zaman	f4
Etkileşimler sonucu çıkan ağırlık ortalaması	f5
Etkileşimler sonucu çıkan memnuniyet ortalaması	f6

Çalışmada kullanılan ikinci öznitelik kümesi ise, tavsiye tabanlı öznitelik kümesidir. Bir kullanıcı başka bir kullanıcı ile etkileşime girmeden önce kendi komşularından etkileşime gireceği kullanıcı hakkındaki deneyimlerini almaktadır. Bunun için bir güven sorgusu yapılmakta ve sorgulanan kullanıcıya ilişkin komşulardan bilgi toplanmaktadır. Komşular tarafından verilen deneyim bilgileri, tavsiye olarak değerlendirilmekte ve o kullanıcı hakkında bir fikir oluşmasını sağlamaktadır. Bu tavsiyelere dayalı olarak da tavsiye tabanlı öznitelik kümesi oluşturulmuştur. Bu kümede; toplam alınan tavsiye sayısı, komşular ile sorgulanan kullanıcı arasında geçen başarılı etkileşim sayısı ortalaması, komşular ile sorgulanan kullanıcı arasında geçen etkileşimlerin ortalama memnuniyetleri, komşular ile sorgulanan kullanıcı arasında geçen etkileşimlerin ortalama ağırlıkları ve komşuların sorgulanan kullanıcıya verdikleri güven metrik değerinin ortalaması bulunmaktadır. Tavsiye tabanlı öznitelik kümesi Çizelge 3 ile verilmiştir.

Çizelge 3: Tavsiye Tabanlı Öznitelikler

Öznitelik	Sembol
Toplam tavsiye sayısı	f7
Tavsiye olarak alınan başarılı etkileşim sayısının ortalaması	f8
Tavsiye olarak alınan memnuniyetlerin ortalaması	f9
Tavsiye olarak alınan ağırlıkların ortalaması	f10
Tavsiye olarak alınan güven metrik değerlerinin ortalaması	f11

5.3.2 Terminal ve Fonksiyon Kümesi

Tez kapsamında geliştirilen genetik modelde problemin çözümüne ulaşmak adına basit fonksiyonlar kullanılmıştır. Fonksiyon kümesi sayesinde önceden belirtilen özniteliklerin yardımı ile bireyler oluşturulmuştur. Bu kümede kullanılan fonksiyonlar ise; toplama, çıkarma, bölme, çarpma, tersini alma, logaritma, karekök ve karesini alma işlemleridir. Fonksiyon kümesi Çizelge 4 ile verilmektedir.

Çizelge 4: Fonksiyon Kümesi

İsim	**Sembol**
toplama	+
çıkarma	-
bölme	/
çarpma	*
tersini alma	1/
logaritma	rlog
karekök	sqrt
karesini alma	square

5.3.3 Uygunluk Fonksiyonu

Uygunluk fonksiyonu, evrimsel hesaplama tekniklerinde başarı performansını oldukça etkileyen bir faktördür. Uygunluk fonksiyonu, genetik programlamanın problemi ne kadar iyi çözeceğini belirlemektedir [67, 33]. Bu tez kapsamında hazırlanan uygunluk fonksiyonu saldırı sayılarındaki azalmayı temel almaktadır. Başka bir değişle eğer güven modeli uygulandığı andaki saldırı sayısına *RguvenVar*, güven modeli olmadan oluşan saldırı sayısına da *RguvenYok* dersek o zaman uygunluk fonksiyonu;

$$\textit{uygunluk fonksiyonu} = R_{guvenVar}/R_{guvenYok} \tag{5.3}$$

Genetik programlama sonucu bulunan çözümler, eğer saldırı sayısını azaltabilirse uygunluk fonksiyonunun değeri düşecektir ve bu da çözümün başarısının yükseldiği anlamına gelmektedir. Yani uygunluk fonksiyonunun sıfıra yaklaşması amaçlanmakta ve bir bireyin başarı ölçütü olarak kullanılmaktadır.

5.4 Eğitim Ve Test Evreleri

Tez kapsamında önerilen modelin başarıya ulaşabilmesi için daha önce anlatılan modülleri kullanarak eğitim ve test evreleri gerçekleştirilmiştir. Eğitim evresi, modelin saldırılara karşı çözüm bulmasını sağlarken, test evresi bulunan çözümün farklı ortamlarda başarısını ölçmektedir. Evrelerin detayına geçmeden önce kullanılan parametreler Çizelge 5 ile gösterilmiştir. Parametreler daha önceden yapılmış çalışmalar referans alınarak oluşturulmuştur [61, 62, 63, 64]. Çizelge 5'te bahsedilmeyen parametreler, ECJ aracının varsayılan parametreleri olarak ele alınmıştır. Gösterilen parametrelerden ise popülasyondaki birey sayısı ve jenerasyon sayısı eğitim evresinde, simülasyon çalıştırma sayısı test evresinde, geri kalan parametreler ise hem eğitim hem de test evresinde kullanılmaktadır.

Çizelge 5: Eğitim ve Test Evresi Parametreleri

Parametre	**Değer**
Popülasyondaki Birey Sayısı	100
Jenerasyon Sayısı	300
Simülasyon Çalıştırma Sayısı	5
Toplam Kullanıcı Sayısı	1000
Toplam Dosya Sayısı	10000
Periyotun Dakika Karşılığı	10
Toplam Periyot Sayısı	50000
Güven Değeri Güncelleme Periyodu	1000

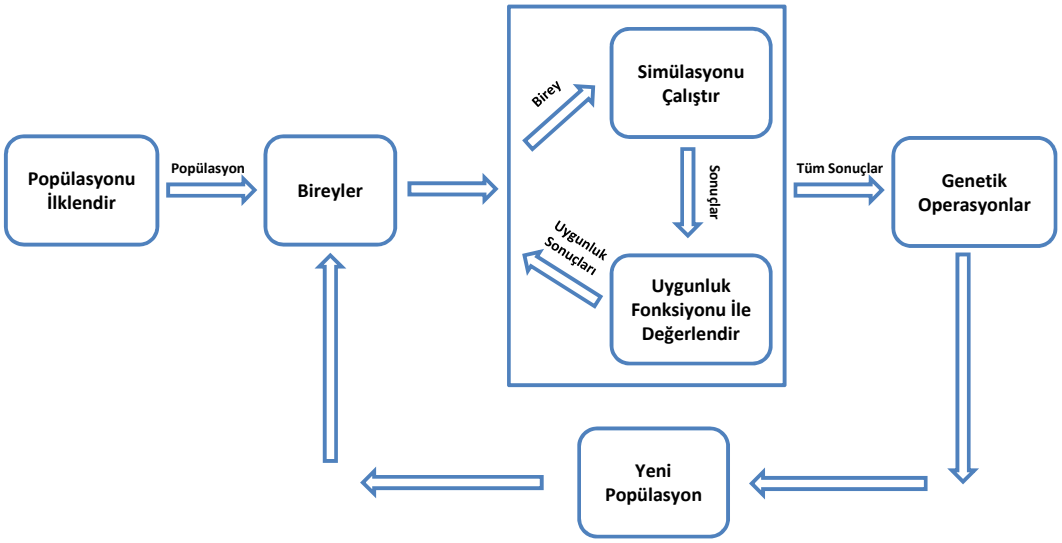
Evrelerde kullanılan parametrelerden de görüldüğü üzere eğitim

evresindeki bir popülasyonda 100 birey bulunurken, eğitim toplam 300 jenerasyon boyunca devam etmektedir. Test evresinde alınan sonuçların doğruluğunu sınamak için aynı test ortamı 5 kere çalıştırılmakta ve çalışmaların sonuçlarının ortalaması modelin başarısı olarak ifade edilmektedir. Ortalamalarının yanında "±" olarak sonuçların standart sapmaları da verilmektedir. Bir ortamda bulunan iyi ve kötü niyetli kullanıcı sayısı 1000 olurken paylaşılan dosya sayısı da 10000 olmaktadır. Simülasyonun çalışması periyotlar ile ifade edilmekte ve bir periyot 10 dakikaya karşılık gelmektedir. Simülasyonun toplam çalışma periyodu ise 50000 periyot olarak belirlenmiştir. Çalışma esnasında güven değerinin güncellenme periyodu ise 1000 periyotta bir olarak düzenlenmiştir.

5.4.1 Eğitim Evresi

Eğitim evresi, temel olarak modelin saldırılara karşı dirençli hale gelmesini sağlamaktadır. Bu evrede genetik programlama modülü ve simülasyon modülü birlikte çalışmaktadır. Genetik programlama modülü, jenerasyondaki her bir birey için simülasyon modülünü baştan sona çalıştırmakta ve sonuçların uygunluk fonksiyonundaki değerlendirilmesine göre yeni jenerasyonlar ile devam etmektedir. Burada simülasyon modülü belirlenen saldırı çeşidine göre devamlı olarak çalışmaktadır. Her bir çalışmada sonuçların daha da iyileşmesi hedeflenmektedir. En son jenerasyona gelindiğinde en iyi sonucun bulunması bu evrenin temel amacıdır. Eğitim evresinin sonunda bulunan çözüm, test aşamasında değerlendirilmektedir.

Çalışmada en önemli nokta, önerilen modelin, eğitim evresinde istenilen saldırı türüne karşı eğitilebilecek olmasıdır. Bu da modele esnek bir yapı ve yeni oluşabilecek durumlara karşı direnç sağlamaktadır. Her zaman yeni saldırı türleri ya da saldırı türü kümeleri modele uygulanabilmekte ve model eğitilerek çözüm bulunabilmektedir.



Şekil 13: Eğitim Evresi

Genetik programlama parametreleri, jenerasyon ve bireyler haricinde ECJ 21 aracının varsayılan parametreleri olarak kullanılmıştır. Eğitim evresinde genel olarak 300 jenerasyon ve 100 birey kullanılmıştır. Her bir saldırı türü için en az 10 eğitim evresi gerçekleştirilmiş ve en iyi sonuçlar test evresinde sınanmıştır. Şekil 13 eğitim evresinin genel adımlarını göstermektedir. Eğitim evresinde önceden bahsedilen tüm öznitelik ve fonksiyon kümeleri kullanılmıştır. Her seferinde farklı sonuçlar verebilecek yapıda olan eğitim evresinin verdiği en iyi sonuçlar test evresinde kullanılmıştır.

5.4.2 Test Evresi

Test evresi, eğitim evresine göre daha basit yapıdadır. Bu evre modelin başarımının ölçüldüğü adımdır. Test evresinde sadece simülasyon modülü kullanılmıştır. Modelin eğitilmesi sonucu bulunan çözüm test evresinde denenmiştir. Sonuçlar test evresinde modelin saldırılara olan direncini yansıtmaktadır. Farklı senaryolar ve eğitimlerden alınan farklı çözümler bu aşamada kullanılmıştır. Çalışmada kullanılan tüm sonuçlar ve istatistikler bu evrenin sonuçlarına göre hazırlanmıştır.

Simülasyon modülünü kullanan bu evre genel olarak 50000 periyotta

çalışmaktadır. Simülasyonda her bir periyot 10 dakika olarak değerlendirilmektedir. Her bir çalışmada normal şartlarda 1000 kullanıcı sistemde bulunmaktadır. Kullanıcılar rastgele olarak dosya indirmekte ve yüklemektedirler. Kullanıcıların da belirli bir yüzdesi iyi kullanıcılardan oluşurken kalanlar kötü kullanıcıları temsil etmektedirler. Test evresinde bulunan sonuçlar, doğruluklarının sınanması ve hata payının ele alınması için aynı test ortamının 5 kere çalıştırılması sonucu elde edilen verilerin ortalaması alınarak hesaplanmaktadır. Ortalamalarının yanında "±" olarak sonuçların standart sapmaları da verilmektedir. Bu evrede yapılan işlemler her 1000 periyotta bir raporlanmakta ve modelin başarısı, saldırıları engelleme sayısı vs. bu raporlama sonucunda elde edilmektedir.

6 DENEYLER VE ÇALIŞMALAR

Bu bölümde modelin test edilmesi için hazırlanan saldırı modelleri ve çalışmalar sonuçları ile birlikte verilecektir.

6.1 Saldırı Türleri

Eşler arası sistemler bünyesinde iyi niyetli ve kötü niyetli kullanıcılar barındırmaktadır. Kötü niyetli kullanıcılar sisteme zarar vermeyi ve kendi yaşamlarını olabildiğince uzun sürdürmeyi hedeflemektedirler. Başlangıçta herkese eşit davranan eşler arası sistemlerin saldırılara açık yapısından dolayı birçok saldırı türüne hedef olabilmektedirler. Tez kapsamında önerilen modelin başarımını görmek için bu saldırı türlerinin bir kısmı kullanılmış ve kullanılan saldırgan türleri bu bölümde açıklanmıştır.

Çalışmada saldırganlar, bireysel ve işbirlikçi olmak üzere iki türde ele alınmıştır. Bu türlerden herhangi birine ait saldırgan ya dosya tabanlı ya da tavsiye tabanlı saldırı yapabilmektedir. Dosya tabanlı saldırılar virüslü ya da içeriği bozuk dosyaların paylaşılması şeklinde yapılan saldırılardır. Tavsiye tabanlı saldırılar ile iki şekilde olabilir. Bunlardan ilki kötü niyetli bir kullanıcının iyi niyetli bir kullanıcı hakkında kötü tavsiye vererek iyi niyetli kullanıcının güven değerini düşürmesidir. İkinci tür tavsiye tabanlı saldırı ise kötü niyetli bir kullanıcının, onunla aynı gruptan olan başka bir kötü niyetli kullanıcı hakkında iyi tavsiyeler vermesidir. Bu sayede kötü niyetli grup arkadaşının güvenini arttırıp iyi niyetli kullanıcılara saldırıda bulunma ihtimalini yükseltmektedir.

Öncelikle sistemde tekil olarak var olan bireysel (*individual*) saldırgan türleri denenmiştir. Bu saldırgan türleri varlıklarını tek başına sürdürmeye çalışmaktadırlar. Bazen birbirlerine bile saldırıda bulunabilirler. Bu saldırganlar iki türde ele alınmıştır:

- **Saf (*Naive*) Saldırganlar:** Bu saldırgan türü her zaman virüslü veya içeriği bozuk dosyalar yüklerken, aynı zamanda her zaman kötü niyetli tavsiye vermektedirler [68].
- **İkiyüzlü (*Hypocritical*) Saldırganlar:** İkiyüzlü saldırganlar %x olasılıkla virüslü veya içeriği bozuk dosya yüklerler ve kötü niyetli tavsiyede bulunurlar[9, 7]. Diğer zamanlarda ise iyi niyetli davranış gösterirler.

Bireysel saldırıların yanı sıra saldırganlar bir takım oluşturup saldırılar yapabilmektedirler. Bu tür saldırganlara işbirlikçi (*collaborators*) saldırganlar denilmektedir. İşbirlikçi saldırganlar birbirlerini bilmekte ve saldırırken koordineli hareket etmektedirler. Kendi aralarında her zaman virüssüz dosyalar paylaşırlar ve birbirlerine iyi tavsiyeler verirler. İşbirlikçi bir saldırgandan başka bir işbirlikçisi hakkında tavsiye istendiğinde her zaman iyi tavsiyeler vererek işbirlikçisinin itibarını arttırmaya çalışmaktadır. Bu durumların dışında işbirlikçi saldırganların davranışları iki şekilde ele alınmıştır:

- **Saf (*Naive*) Saldırgan İşbirlikçiler:** Bu türdeki işbirlikçiler, iyi niyetli ve işbirlikçisi olmayan kullanıcılara her zaman virüslü veya içeriği bozuk dosyalar yüklerken aynı zamanda her zaman kötü niyetli tavsiye vermektedirler [68].
- **İkiyüzlü (*Hypocritical*) Saldırgan İşbirlikçiler:** İkiyüzlü saldırganlar %x olasılıkla iyi niyetli ve işbirlikçisi olmayan kullanıcılara virüslü veya içeriği bozuk dosya yüklerler ve kötü niyetli tavsiyede bulunurlar[9, 7]. Diğer zamanlarda ise iyi niyetli davranış gösterirler.

Bunların dışında son olarak kimlik değiştiren (*pseudospoofers*) saldırganlar da modelde denenmiştir [69, 7]. Kimlik değiştiren saldırganlar bazı zamanlarda kimliklerini değiştirerek saldırgan davranışlarını

gizlemektedirler. Böylece saldırganların sistemdeki varlıkları daha uzun ömürlü olur.

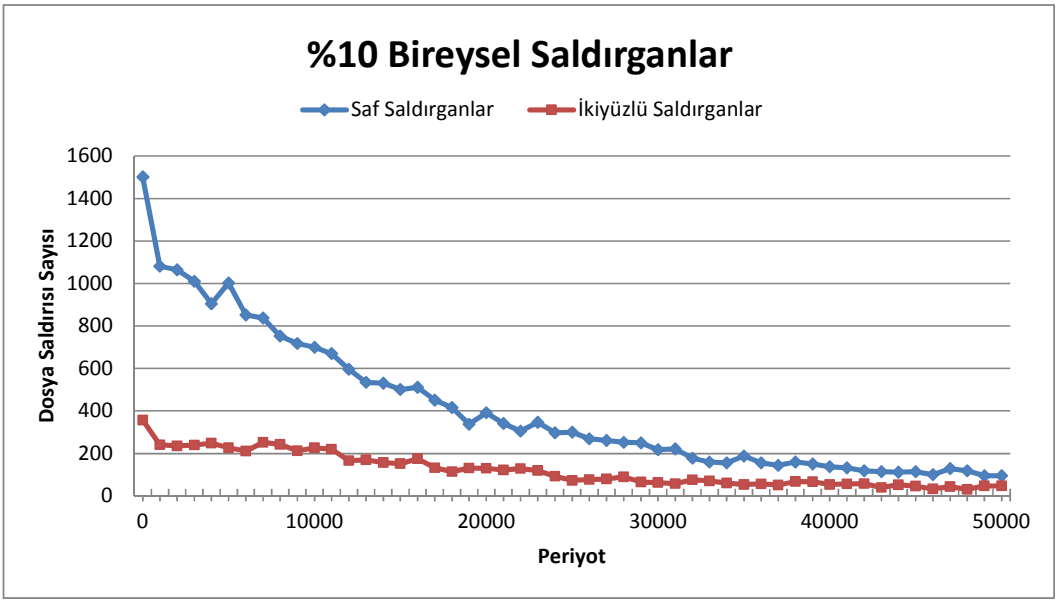
6.2 Bireysel Saldırganlar

Bireysel saldırılar sistemde bir topluluğa üye olmadan tekil olarak var olan saldırganların yaptığı saldırıları içermektedir. Bu deneyde hem saf saldırganlar hem de ikiyüzlü saldırganlar için kullanıcıların %10'u saldırgan olacak şekilde hazırlanmış bir ortamda model eğitilmiştir. 10 farklı ortamda yapılmış eğitimlerden ise en başarılı sonuç vereni test edilmiştir. Eğitilmiş model %10, %30 ve %50 saldırgan oranlarının olduğu ortamlarda denenmiştir. Bu deneyde ikiyüzlü saldırganların saldırı ihtimali %20 olarak alınmıştır.

Çizelge 6: Bireysel Saldırganlarda Dosya Tabanlı Saldırıların Engellenme Oranları

	%10	**%30**	**%50**
Saf Saldırganlar	83.8 ± 1.3	78.9 ± 1.6	73.6 ± 0.7
İkiyüzlü Saldırganlar	71.8 ± 1.1	57.7 ± 0.9	47.1 ± 1.5

Çizelge 6, dosya tabanlı bireysel saldırılara karşı modelin saldırıları engelleme oranlarını göstermektedir. Buradaki yüzdeler, güven modeli içermeyen bir sistemdeki saldırı sayısının, önerilen model sayesinde yüzde kaç oranında azaltıldığını ifade etmektedir. Sonuçlara bakıldığında, modelin saf saldırganlar üzerinde oldukça yüksek bir başarı oranına sahip olduğu görülmektedir. Bunun sebebi, saf saldırganların devamlı saldırı halinde olması ve tespit edilmesi kolay saldırganlar olmalarıdır. İyi niyetli bir kullanıcı ile girdikleri ilk etkileşimde yaptıkları saldırı sebebiyle hemen fark edilmektedirler. Belirli bir olasılıkla saldıran ve saf saldırganlara göre zor bir saldırgan türü olan ikiyüzlü saldırganlarda da model yine %71,8'lik kayda değer bir başarı elde etmiştir. İkiyüzlü saldırganların zaman zaman iyi

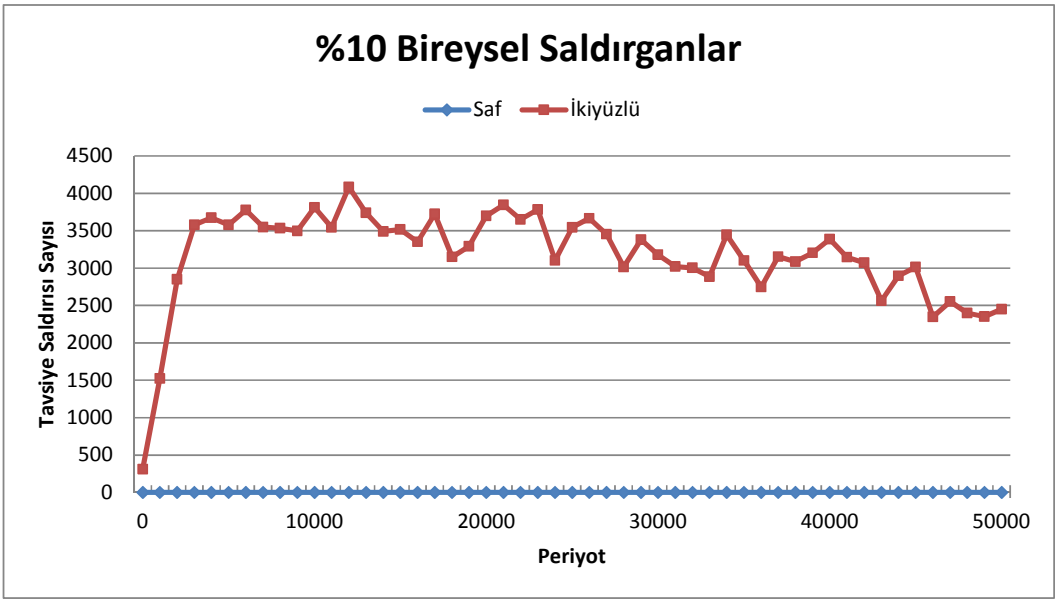


Şekil 14: %10 Bireysel Saldırgan İçeren Ortamdaki Zamana Bağlı Dosya Tabanlı Saldırı Sayıları

niyetli kullanıcı gibi davranmaları ve güven değerlerini arttırmaları sebebiyle saf saldırganlar kadar yüksek başarı elde edilememiştir. Fakat elde edilen sonuç bu tür zor saldırganlara karşı yüksek bir başarı göstergesidir. Modelin, saldırganın saldırı yapısına göre eğitilebilir olması ve saldırganı iyi tanıması bu başarıyı getirmiştir. Diğer taraftan %50 gibi yüksek oranda saldırgan barındıran bir ortamda da yine ikiyüzlü saldırganlara karşı saldırıların neredeyse yarısını engellenebilmiştir. Bu da yoğun saldırgan içeren bir ortamda sistemin başarısını göstermektedir.

Bir güven modelinin, sistemde oluşan saldırı sayısını zamana bağlı olarak düşürme yeteneği modelin başarısını göstermektedir. Şekil 14 %10 saldırganın bulunduğu bir ortamdaki dosya tabanlı saldırı sayılarının zamana bağlı değişimini göstermektedir. Her 1000 periyotta bir örneklem alınmıştır. Saldırılar fazla direnç olmadan zaman içerisinde hızla azalmaktadır.

Şekil 15, %10 saldırgan içeren bir ortamda tavsiye tabanlı saldırıların her 1000 periyottaki grafiğini göstermektedir. Modelde tavsiyeler daha önce güven kazanılmış komşulardan alınmaktadır. Saf saldırganlar devamlı



Şekil 15: %10 Bireysel Saldırgan İçeren Ortamdaki Zamana Bağlı Tavsiye Tabanlı Saldırı Sayıları

saldırı yaptıklarından ve başarılı bir etkileşime giremediklerinden dolayı hemen fark edilmekte ve kimse ile güvenilir komşu olamamaktadırlar. Bu yüzden, saf saldırganlar için tavsiye tabanlı saldırıların sayısı hiçbir zaman kimseyle komşu olup tavsiye veremediklerinden dolayı sıfır olmaktadır. Fakat aynı şey belirli bir yüzde ile saldıran ikiyüzlü saldırganlar için geçerli değildir. İkiyüzlü saldırganlar, iyi niyetli kullanıcılar ile virüssüz ya da bozuk içeriğe sahip olmayan dosya paylaşmasının ardından başarılı bir etkileşim gerçekleştirerek kullanıcıların komşusu oldukları için başka etkileşimler sırasında rahatlıkla kötü niyetli tavsiyeler verebilmektedirler. Saldırı sayısının dosya tabanlı saldırılar gibi hemen düşmemesinin sebebi ise tavsiye tabanlı saldırıların dosya tabanlı saldırılar kadar kolay fark edilebilir yapısının olmamasıdır. Model, belirli bir süre içerisinde tavsiye tabanlı saldırıları belirli bir sınırda tutmakta ve zamanla ikiyüzlü saldırganlar tanındıkça saldırıları azaltmaktadır.

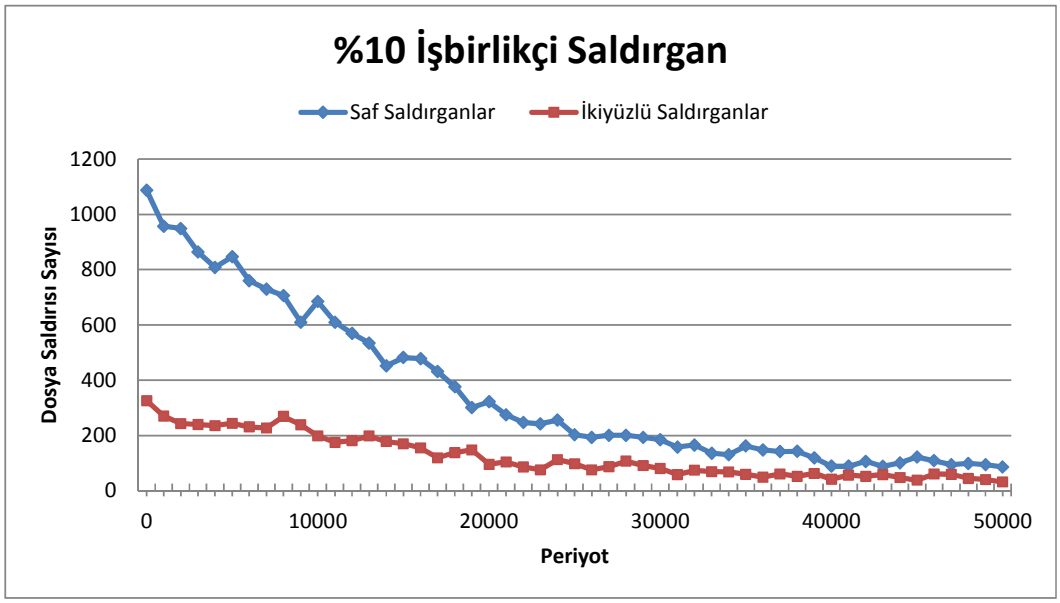
6.3 İşbirlikçi Saldırganlar

İşbirlikçi saldırganlar tekil hareket etmek yerine grup halinde saldırılar düzenleyerek sistemde varlıklarını daha uzun ömürlü tutmaya çalışmaktadırlar. Deneylerin bu aşamasında bireysel saldırganlara göre daha zor olan işbirlikçi saldırganlar üzerine çalışılmıştır. Model hem saf hem de ikiyüzlü işbirlikçi saldırganlar için %10 saldırgan oranı içeren ortamda eğitilmiş ve yapılan 10 tane eğitimden en iyi sonuç vereni test edilmiştir. Burada bir işbirlikçi grubun maksimum saldırgan sayısı 50 olarak alınmıştır. Bu da ortamda farklı saldırgan işbirlikçi grupların olmasını ve kendi gruplarından olmayan diğer saldırganlara da saldırmalarını sağlamıştır. Testler %10, %30 ve %50 saldırgan içeren ortamlarda gerçekleştirilmiş ve ikiyüzlü saldırganların saldırı olasılığı %20 olarak alınmıştır. Sonuçlar bireysel saldırganlarda olduğu gibi dosya tabanlı saldırılar ve tavsiye tabanlı saldırılar olarak değerlendirilmiştir.

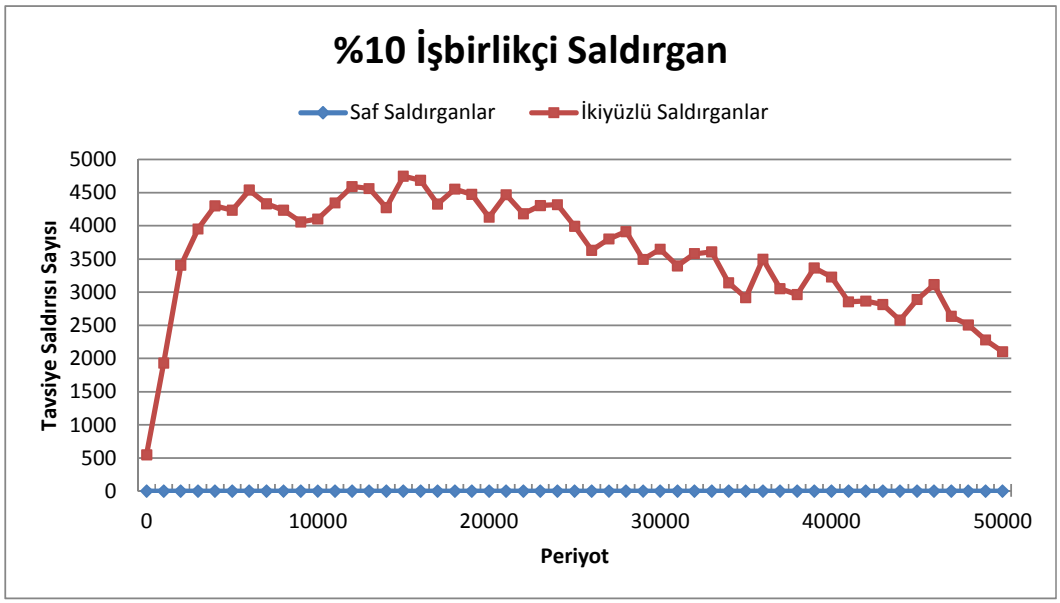
Çizelge 7: İşbirlikçi Saldırganlarda Dosya Tabanlı Saldırıların Engellenme Oranları

	%10	**%30**	**%50**
Saf Saldırganlar	79.3 ± 0.9	75.1 ± 1.2	71.9 ± 0.7
İkiyüzlü Saldırganlar	61.7 ± 1.2	46.3 ± 1.1	39.5 ± 1.4

Çizelge 7, modelin işbirlikçi saldırganlara karşı olan başarı yüzdelerini göstermektedir. Basit bir saldırgan türü olan saf saldırganlar işbirlikçi olmanın avatajlarından fazla yararlanamamışlardır. Bunun sebebi, ilk etkileşimlerinde doğrudan saldırmalarından dolayı tespit edilmesi kolay olan saldırganların ekip olarak da fazla bir başarı gösterememeleridir. İlk etkileşimden sonra kimlikleri belirlenen saf saldırganlar o anda sistemden dışlanmaktadır. Saf saldırganlardaki %10 saldırgan olan ortamda başarı yüzdesi %79,3 iken, %50 oranında saldırgan olan ortamda bu oran ancak %71,9'a düşmektedir. Bu da yine bireysel saldırganlarda da olduğu



Şekil 16: %10 İşbirlikçi Saldırgan İçeren Ortamdaki Zamana Bağlı Dosya Tabanlı Saldırı Sayıları



Şekil 17: %10 İşbirlikçi Saldırgan İçeren Ortamdaki Zamana Bağlı Tavsiye Tabanlı Saldırı Sayıları

gibi yüksek bir başarıyı temsil etmektedir. Saf saldırganlara karşın ikiyüzlü saldırganlar işbirlikçi olmanın faydasını görmektedirler. Belirli bir olasılıkla saldıran ikiyüzlü saldırganlar tespit edilmeden önce kendi işbirlikçileri hakkında iyi tavsiyeler vererek sistemdeki varlıklarını uzatmakta ve tespit edilmelerini zorlaştırmaktadırlar. Bu sayede hem birbirlerinin güven değerini artırmakta hem de iyi niyetli kullanıcılara daha fazla saldırıda bulunabilmektedirler. İşbirlikçi yapılarına karşı %10 saldırgan içeren bir ortamda model %61,7'lik bir başarı oranı yakalamaktadır. Hem saf hem de ikiyüzlü saldırganların %50 saldırgan içeren ortamdaki saldırılarına karşı modelin gösterdiği başarının saf saldırganlarda çok fazla düşmediği fakat ikiyüzlü saldırganlarda yaklaşık %20 oranında düşüş olduğu görülmektedir. İki tür saldırgan arasındaki bu fark, saf saldırganların sürekli ve yoğun saldırı yapmalarından dolayı hemen fark edilmelerinden kaynaklanmaktadır. İkiyüzlü saldırganlar ise yoğun saldırgan bulunan ortamda yaptıkları iyi niyetli etkileşimler sayesinde itibarlarını daha fazla arttırdıkları için fark edilmeleri zor olmaktadır. Bu yüzden ortamdaki saldırgan yoğunluğu %50'ye geldiğinde, iki saldırgan türü arasında modelin başarısı açısından fark oluşmaktadır.

Şekil 16, %10 işbirlikçi saldırgan olan ortamdaki dosya tabanlı saldırıların her 1000 periyottaki miktarı göstermektedir. Saldırı sayıları işbirlikçi yapılarına rağmen hızla azalmaktadır.

İşbirlikçi saldırganların %10 oranında ortamda bulunması sonucu yaptığı tavsiye tabanlı saldırıları Şekil 17 göstermektedir. Saf saldırganlar ilk etkileşimde tespit edildiklerinden dolayı tavsiye tabanlı saldırı gerçekleştirememişlerdir. Fakat ikiyüzlü saldırganlar işbirlikçilerinin de etkisiyle bireysel davranışlarına göre tavsiye tabanlı saldırı sayılarını arttırmışlardır. Bu artışa rağmen model zamana bağlı olarak tavsiye tabanlı saldırıların sayısını başarılı bir şekilde azaltmıştır.

6.4 Kimlik Değiştiren Saldırganlar

Kimlik değiştiren saldırganların belirli aralıklarla sistemde varlıklarını sürdürebilmek için komşuluklarını bozan ve sisteme yeni girmiş kullanıcı gibi davranan karakteristikleri vardır. Tez kapsamında geliştirilen model bu saldırgan türlerine göre de %10 kimlik değiştiren saldırgan içeren ortamda eğitilmiştir. Yapılan 10 farklı eğitimden en iyi sonuç vereni alınmış ve test edilmiştir. Testler %10, %30 ve %50 saldırgan içeren ortamlarda gerçekleştirilmiştir. Testler sırasında ikiyüzlü saldırganların saldırı olasılığı %20 alınmıştır. Kimlik değiştiren saldırganlar ise her 1000 periyotta bir kimliklerini değiştirmektedirler.

6.4.1 Bireysel Kimlik Değiştiren Saldırganlar

Çizelge 8: Kimlik Değiştiren Bireysel Saldırganlarda Dosya Tabanlı Saldırıların Engellenme Oranları

	%10	**%30**	**%50**
Saf Saldırganlar	51.2 ± 0.6	47.5 ± 1.0	41.1 ± 0.4
İkiyüzlü Saldırganlar	53.9 ± 0.8	46.1 ± 1.2	36.6 ± 0.7

Testler öncelikle kimlik değiştiren bireysel saldırganlar üzerinden yapılmıştır. Çizelge 8 kimlik değiştiren bireysel saf ve ikiyüzlü saldırganlara karşı modelin farklı saldırgan oranlarındaki ortamlarda elde ettiği başarı yüzdelerini göstermektedir. Saf saldırganlara karşı model %51,2'lik bir başarı gösterirken, ikiyüzlü saldırganlara karşı %53,9'luk bir başarı sergilemiştir. Saf saldırganların normal davranışı aksine kimlik değiştiren yapıda davranması üzerine modelin başarısı düşmüştür. Bunun sebebi sürekli saldırma davranışında olan saf saldırganların normal davranışlarında hemen yakalanmalarından dolayı modelin başarısı yüksek çıkarken, kimlik değiştiren yapıda saldırmalarında yakalansalar bile

hemen kimliklerini değiştirip kötü etkileşim geçmişlerini temizlemelerinden ve tekrar saldırmalarından dolayı başarıyı düşürmeleridir. Bu durum ikiyüzlü saldırganlar için biraz farklıdır. İkiyüzlü saldırganların en büyük güçleri olan, iyi niyetli kullanıcılar ile girdikleri başarılı etkileşimler sayesinde edindikleri güveni kaybetmelerinden dolayı modelin başarısı saf saldırganlara göre yüksek çıkmaktadır. İyi niyetli kullanıcılarla yaptıkları başarılı etkileşimler sayesinde elde ettikleri güvenilirliklerini kimlik değiştirme esnasında kaybeden ve yeni kimlikleri ile sistemde tanınmamasından dolayı başarıları düşen ikiyüzlü saldırganlar bu saldırı şeklinden çok fazla fayda sağlayamamışlardır. Model %50 kimlik değiştiren saldırgan içeren ortamda da dikkate değer bir başarı göstermektedir.

6.4.2 İşbirlikçi Kimlik Değiştiren Saldırganlar

Kimlik değiştiren saldırganlar için yapılan testlerin ikinci aşamasında bireysel saldırganlar yerine işbirlikçi saldırganlar kullanılmıştır. İşbirlikçi saldırganlarda bir gruba üye saldırgan sayısı maksimum 50 olarak alınmıştır.

Çizelge 9: Kimlik Değiştiren İşbirlikçi Saldırganlarda Dosya Tabanlı Saldırıların Engellenme Oranları

	%10	**%30**	**%50**
Saf Saldırganlar	50.4 ± 1.1	46.7 ± 0.6	37.9 ± 0.7
İkiyüzlü Saldırganlar	51.3 ± 1.0	44.2 ± 0.3	31.7 ± 1.2

Çizelge 9, kimlik değiştiren işbirlikçi saldırganlara karşı modelin başarı yüzdelerini göstermektedir. İşbirlikçi saldırganlara karşı farklı saldırgan oranlarındaki saf ve ikiyüzlü saldırganlar %10 saldırgan oranındaki ortamda eğitilmesi sonrasında test edilmiştir. Sonuçlarda görüldüğü üzere kimlik değiştiren işbirlikçi saf saldırganlara karşı %50,4'lük başarı elde edilirken, ikiyüzlü saldırganlara karşı ise %51,3'lük bir başarı elde edilmiştir. Saf saldırganlar normal davranışlarında da işbirlikçi olmanın avantajından

yararlanamadıkları gibi kimlik değiştirme davranışlarında da işbirlikçi olmaktan fazla bir fayda sağlayamamışlardır. Burada modelin kimlik değiştiren işbirlikçi saf saldırganlar karşısında düşük başarı göstermesinin tek sebebi kolay tespit edilen saf saldırganların kimliklerini değiştirerek tekrar sisteme dâhil olmaları ve saldırıda bulunabilmeleridir. İkiyüzlü saldırganlar da kimlik değiştirmelerinden dolayı diğer kullanıcılara karşı olan itibarlarını kaybetmeleri sebebiyle kimlik değiştirmelerinden ve işbirlikçi avantajlarından çok fazla yararlanamamışlardır. Diğer kullanıcıların gözündeki itibarlarını yeni kimliklerinde bulamayan ikiyüzlü saldırganlar bu değişimden çok fazla fayda sağlayamamışlardır. Model, her iki saldırgan türüne karşı da başarı göstermiştir. Fakat saf saldırganlar ikiyüzlü saldırganlara nazaran kimlik değiştirme davranışlarından daha fazla fayda görmüşlerdir.

6.5 Karışık Orandaki Saldırganlar

Her bir saldırgan türünde ayrı ayrı yapılan testlerin ardından modelin karışık saldırgan kümesi üzerindeki başarısı da denenmiştir. Farklı yoğunlukta ve farklı türlerde saldırgan içeren ortamlarda model eğitilmiş ve test edilmiştir. Eğitimler karışık saldırgan türlerinden %30 oranında saldırgan içeren ortamda gerçekleştirilmiştir. Eğitimlerin sonucunda testler %30 ve %50 saldırgan içeren ortamda test edilmiştir. Bahsi geçen karışık türde saldırganları içeren ortamda;

- Bireysel saf saldırganlar,
- Bireysel ikiyüzlü saldırganlar,
- İşbirlikçi saf saldırganlar,
- İşbirlikçi ikiyüzlü saldırganlar,
- Kimlik değiştiren bireysel saf saldırganlar,

- Kimlik değiştiren bireysel ikiyüzlü saldırganlar,
- Kimlik değiştiren işbirlikçi saf saldırganlar,
- Kimlik değiştiren işbirlikçi ikiyüzlü saldırganlar

bulunmaktadır.

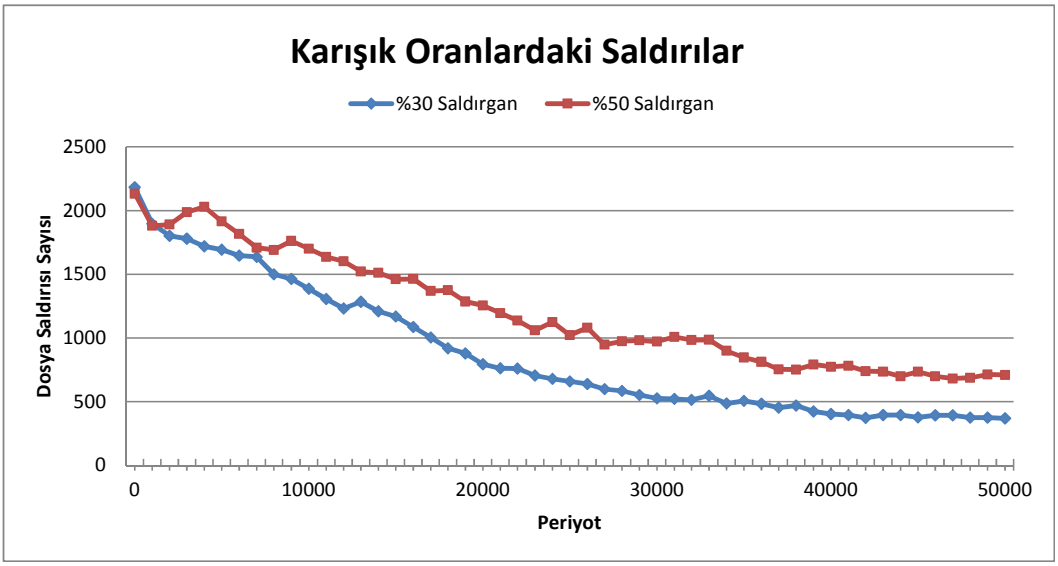
Yapılan testler sırasında işbirlikçi bir grubun saldırgan sayısı 50, ikiyüzlü saldırganların saldırı olasılığı %20 ve kimlik değiştirme periyodu 1000 periyot olarak alınmıştır.

Çizelge 10: Karışık Saldırgan Bulunan Ortamdaki Dosya Tabanlı Saldırıların Engellenme Oranları

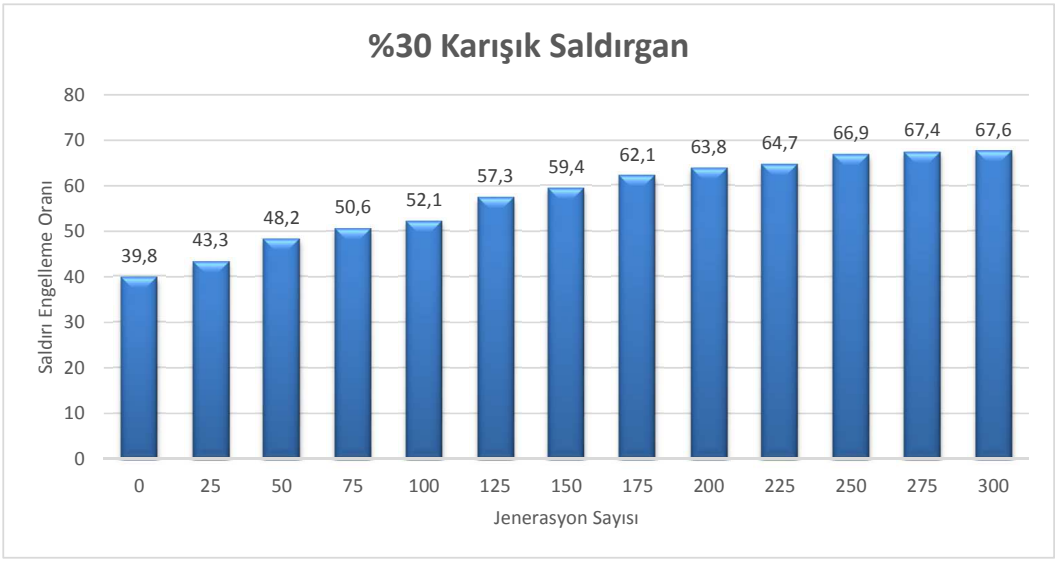
	%30	**%50**
Karışık Saldırganlar	67.6 ± 0.4	53.3 ± 1.2

Farklı yoğunlukta saldırgan içeren ortamlara karşı modelin elde ettiği başarı oranları Çizelge 10 ile verilmiştir. Sonuçların da gösterdiği üzere %30 oranında saldırgan içeren bir ortamda %67,6'lık ve %50 oranında saldırgan içeren ortamda ise %53,3'lük yüksek bir başarı göstermiştir. Farklı saldırgan karakteristiklerine sahip kullanıcıların eşler arası sistemlerde var olması gibi gerçek dünyaya yakın olan bu senaryolardaki başarısı modelin gerçekçi yaklaşımını ve kabul edilir sonuçlar verdiğini göstermiştir. Birçok saldırgan türünü bulunduran ve saldırganların yüksek oranda var olduğu ortamlarda bile kendini geliştiren model sayesinde hafife alınmayacak başarılar elde edilmiştir. Tüm saldırgan türlerinin bulunduğu ortamlarda da ortak bir çözüm bulma eğiliminde olan model sayesinde saldırılar engellenebilmiştir.

Şekil 18, saldırganlar tarafından gerçekleştirilen saldırıların her 1000 periyottaki sayıları göstermektedir. İlk başta davranışları ve yoğunlukları sebebiyle bulunmakta zorlanılan saldırganlar, zamanla iyi kullanıcılar tarafından tespit edilerek izole edilmiş ve buna bağlı olarak saldırı sayıları hızla azalmıştır.



Şekil 18: Karışık Oranda Saldırgan İçeren Ortamda Zamana Bağlı Dosya Tabanlı Saldırı Sayıları



Şekil 19: %30 Saldırgan İçeren Ortamdaki Jenerasyon Sayısına Bağlı Dosya Tabanlı Saldırı Engelleme Oranları

Şekil 19, modelin %30 oranında karışık saldırgan içeren ortamda yapılan eğitim esnasındaki saldırıları engelleme oranlarının jenerasyon sayısına bağlı olarak gelişimi gösterilmiştir. Her 25 jenerasyonda bir gösterilen değerlerde, modelin zaman içerisinde saldırılara karşı olan başarı artışı görülmektedir. Bulduğu çözümleri zamanla genetik operasyonlara tabi tutan model, bulunan çözümleri giderek iyileştirmektedir. Gösterilen değerler bulunulan jenerasyondaki en başarılı sonucu içermektedir. Bazı durumlarda bulunulan sonuçlar bir öncekinden daha düşük başarıda ya da aynı başarıda olabilmektedir. Fakat şekilde o jenerasyona kadar elde ettiği en başarılı sonuç verilmiştir.

6.6 Ağırlık Ve Memnuniyet Özniteliklerinin Etkisi

Genetik modelin öznitelik kümesinde hem tavsiye ve hem de etkileşim tabanlı olarak elde edilen ağırlık ve memnuniyet değerlerinin saldırı tespitine etkisi yapılan eğitim ve testler ile ölçülmeye çalışılmıştır. Eğitim ve test işlemlerinde saldırgan türü olarak bireysel saf saldırganlar, bireysel ikiyüzlü saldırganlar, işbirlikçi saf saldırganlar ve işbirlikçi ikiyüzlü saldırganlar kullanılmıştır. Bu kapsamda önce ağırlık değeri öznitelik kümesinden çıkartılmış ve her bir saldırgan türü için ayrı ayrı eğitimler gerçekleştirilmiştir. Elde edilen en iyi sonuç ile hem bireysel hem de işbirlikçi saf ve ikiyüzlü saldırganlar üzerinde test edilmiştir. Ardından memnuniyet değeri öznitelik kümesinden çıkartılıp bahsi geçen saldırgan türleri için model ayrı ayrı eğitilmiş ve ağırlık özniteliği testinde olduğu gibi aynı saldırgan türleri ile test edilmiştir. Eğitimler ve testler %10 saldırgan içeren ortamlarda gerçekleştirilmiştir. Testlerde ikiyüzlü saldırganların saldırı olasılığı %20, işbirlikçi grubun saldırgan sayısı ise 50 olarak alınmıştır.

Çizelge 11 ile ağırlık ve memnuniyet değerlerinin ayrı ayrı çıkarılması sonucu modelin belirtilen saldırgan türlerine karşı elde ettiği başarı oranları verilmiştir. Bireysel saf saldırganlarda memnuniyet olmadığı zaman %75,9,

Çizelge 11: Memnuniyet Ve Ağırlık Özniteliklerinin Dosya Tabanlı Saldırıları Engellemeye Etkisi

	Memnuniyet Yok	**Ağırlık Yok**	**Hepsi Var**
Bireysel Saf	75.9 ± 0.5	76.4 ± 1.1	83.8 ± 1.3
Bireysel İkiyüzlü	55.1 ± 0.2	58.2 ± 0.7	71.8 ± 1.1
İşbirlikçi Saf	70.8 ± 1.4	71.2 ± 0.8	79.3 ± 0.9
İşbirlikçi İkiyüzlü	49.4 ± 0.6	52.1 ± 1.3	61.7 ± 1.2

ağırlık olmadığı zaman ise %76,4'lük başarı elde edilmiştir. Bu bize basit saldırgan türlerinin tespit edilmesi kolay olduğu için ikisinden birinin yokluğunun başarı oranını çok düşürmediğini göstermektedir. Çıkarılan öznitelik dışındakiler ile başarıda düşüş olsa da kabul edilebilir bir başarı elde edebilmektedir. Saldırının karmaşık bir yapıda olmaması ve kolay fark edilmesi sebebiyle bu sonuçlar alınabilmiştir. İkiyüzlü saf saldırganların ise hem bireysel hem de işbirlikçi saldırı sonuçlarına bakıldığında memnuniyet ya da ağırlık özniteliklerinin çıkarılmasının normal eğitimlere göre kıyaslandığında başarı oranını daha fazla düşürdüğü görülmüştür. Bu da bize karmaşık ve zor saldırgan türlerinde öznitelik kümesinde bulunan memnuniyet ve ağırlık değerlerinin önemini göstermiştir. Kullanılan öznitelik kümesinin çeşitliliği ve anlam bakımdan gücü, modelin başarısını zorlu saldırganlar üzerinde doğrudan etkilemiştir. Modelin eğitiminde kullanılan öznitelik kümesinin çeşitliği ve etkinliği zor saldırganlar üzerinde daha da önem kazanmaktadır.

Bulunan sonuçlara ek olarak, yapılan testler sonucunda memnuniyetin ağırlığa göre başarıyı daha fazla etkilediği sonucu çıkartılmıştır. Bu da bize modelin eğitiminde memnuniyet değerinin ağırlığa nazaran daha önemli olduğunu göstermektedir. Yapılan etkileşimlerin başarılı ya da başarısız olmaları doğrudan memnuniyeti etkilediğinden dolayı memnuniyet değerleri modelin başarısını daha da etkilemiştir.

6.7 Saldırganlar Arasında Çapraz Eğitim Ve Testler

Bundan önce yapılan testlerde modelin eğitimleri belirli bir saldırgan türü ya da saldırgan kümesi üzerinde yapılarak aynı saldırgan kümesi ya da saldırı türü üzerinden test edilmiştir. Bu adımda modelin eğitilebilirliğinin ve saldırgan davranışlarına göre şekil alabilmesinin test edilmesi için eğitim ve test adımları saldırgan türleri arasında çapraz olarak gerçekleştirilmiştir. Bireysel saf ve ikiyüzlü saldırganların eğitimleri sonucunda elde edilen çözümler ayrı ayrı hem saf hem de ikiyüzlü saldırgan ortamlarında test edilmiştir. Eğitim ve testlerde %10 saldırgan içeren ortamlar kullanılmıştır. İkiyüzlü saldırganların saldırı olasılığı %20 olarak alınmış ve yapılan 5 testin ortalama başarıları sonuç olarak verilmiştir.

Çizelge 12: Bireysel Saf-İkiyüzlü Saldırganların Çapraz Eğitim Ve Testlerinin Başarı Oranları

		TEST	
		Saf Saldırgan	**İkiyüzlü Saldırgan**
EĞİTİM	**Saf Saldırgan**	83.8 ± 1.3	55.9 ± 0.6
	İkiyüzlü Saldırgan	81.4 ± 0.8	71.8 ± 1.1

Çizelge 12 ile eğitim ve testlerin saldırgan türlerine göre çapraz olarak yapılmış sonuçları verilmiştir. Saf saldırgan türüne karşı eğitilen model saf saldırganlar üzerinde test edildiğinde, daha önce de verildiği gibi %83,8'lik bir başarı elde etmiştir. Aynı şekilde ikiyüzlü saldırgan türüne göre eğitilen model yine ikiyüzlü saldırganlara karşı yapılan testler esnasında %71,8'lik bir başarı sağlamıştır. Testin amacı olarak alınan bir türde eğitilen modelin başka bir türde denenmesi sonucu, saf saldırganlarla eğitilmiş bir model ikiyüzlü saldırganlar üzerinde test edildiğinde %55,9'luk bir başarı göstermiştir. Normalde ikiyüzlü saldırganlar ile eğitilip yine ikiyüzlü saldırganlar ile test edilerek sağlanan başarıdaki yaklaşık %16'lık bu düşüş, saf saldırganlar gibi basit saldırgan türlerinde eğitilen modelin, zor

saldırgan türlerinde aynı başarıyı yakalayamadığını göstermektedir. Çünkü saf saldırgan türüne göre eğitilen model, çözümü basit olan bir probleme basit bir çözüm bulmaktadır. Fakat zorlu bir saldırgan türünde bu çözüm yeterince etkili ve güçlü olamamaktadır. Modelin zor bir saldırgan türü olan ikiyüzlü saldırganlara göre eğitilip kolay bir saldırgan olan saf saldırganlarda test edilmesi sonucu da %81,4'lük bir başarı sağlanmıştır. Normalde saf saldırganlar ile eğitilip test edilen modelin sağladığı %83.8'lik başarıdaki yaşanan ve yok sayılabilecek kadar küçük olan yaklaşık %2'lik bu düşüş, zor bir saldırgana göre eğitilen modelin hem zor saldırganlar üzerinde hem de kolay saldırganlar üzerinde başarılı olduğunu göstermektedir.

Çizelge 13: Bireysel-İşbirlikçi Saldırganların Çapraz Eğitim Ve Testlerinin Başarı Oranları

		TEST	
		Bireysel İkiyüzlü	**İşbirlikçi İkiyüzlü**
EĞİTİM	**Bireysel İkiyüzlü**	71.8 ± 1.1	49.6 ± 0.4
	İşbirlikçi İkiyüzlü	68.9 ± 1.5	61.7 ± 1.2

Çapraz olarak yapılan eğitim ve testlerin ikinci aşamasında, daha zor bir davranış türü olan ve işbirlikçi olarak hareket eden saldırgan türleri ile daha kolay bir davranış türü olan ve bireysel olarak hareket eden saldırgan türleri arasındaki ilişkinin bir önceki adımda bulunan sonuçlar ile uygunluğu araştırılmıştır. Çizelge 13 bireysel ikiyüzlü saldırganlar ile işbirlikçi ikiyüzlü saldırganlar arasında yapılan çapraz eğitim ve testlerin sonuçlarını göstermektedir. Bireysel ikiyüzlü saldırganlar ile eğitilen model, yine bireysel ikiyüzlü saldırganlar üzerinde test edildiğinde %71,8'lik bir oranla saldırıları engellemektedir. İşbirlikçi ikiyüzlü saldırganlar üzerinde eğitilen model, bireysel ikiyüzlü saldırganlar üzerinde test edildiğinde %68,9'luk bir oranla saldırıları engellemektedir. Bu sonuç, zor bir saldırgan türü olan işbirlikçi ikiyüzlü saldırganlarla eğitilen modelin, işbirlikçi saldırganlara göre daha kolay olan bireysel ikiyüzlü saldırganlarla test

edilmesinin, modelin saldırıları engelleme oranını çok fazla düşürmediğini göstermektedir. Yani zor saldırgan türünde eğitilen modelin kolay saldırgan türleri üzerinde de başarılı olduğunu göstermektedir. Diğer taraftan işbirlikçi ikiyüzlü saldırganlara göre eğitilen model, yine işbirlikçi ikiyüzlü saldırganlar üzerinde test edildiğinde %61,7 oranında saldırıları engellemektedir. Fakat işbirlikçi ikiyüzlü saldırganlara göre kolay bir saldırgan türü olan bireysel ikiyüzlü saldırganlara karşı eğitilen model, işbirlikçi ikiyüzlü saldırganlar üzerinde test edildiğinde ise modelin saldırıları engelleme oranı %49,6'ya düşmektedir. Bu da kolay bir saldırgan türüne karşı eğitilmiş modelin, daha zor olan saldırganlara karşı olan başarı oranının düştüğünü göstermektedir.

İki çalışmanın da sonuçları gösteriyor ki modelin saldırılara karşı direnci, saldırıların zorlukları ve karmaşıklıkları ile doğru orantılı olarak artmaktadır. Bu sonuç, modelin saldırının seviyesine göre daha güçlü çözümlere ulaşabildiğini ve daha başarılı sonuçlar verebildiğini göstermektedir.

7 SONUÇ

Eşler arası sistemlerde güven yönetiminin sağlanması, saldırgan türlerinin çeşitliliği ve karakteristiklerinin farklı olması sebebiyle oldukça zor bir problemdir. Bu problemin çözümü için geniş bir arama uzayı bulunmaktadır. Tez kapsamında yapılan çalışmada çözümü zor olan problemler için sonuçları daha da iyiye götürecek şekilde evrimleşecek bir çözüm sunulmuştur. Çalışmada eşler arası sistemlerdeki güven yönetiminin sağlanması için genetik programlama yardımı ile evrimleşebilen bir model geliştirilmiştir. Saldırgan türleri ve davranışlarına göre evrimleşen model sayesinde başarılı sonuçlar sağlanmıştır. Sabit formüller ya da istatistiki yöntemlere nazaran eşler arası sistemlerin yaşadığı güvenlik sorunlara çözüm bulan bu model, eşler arası sistemlerin doğasına daha yakın davranmaktadır.

İlk aşamada modelin testleri için modüller oluşturulmuştur. Simülasyon ve genetik programlamadan oluşan bu modüller, eğitim ve test evreleri için kullanılmaktadır. Öncelikle eşler arası bir sistemi modellemek adına gerçek davranışına çok yakın olan simülasyon modülü [60] çalışması temel alınarak geliştirilmiştir. Bu modül, kullanıcıların birbirleri arasında etkileşimde bulunmalarını ve gerektiği durumlarda tavsiye vermelerini sağlamaktadır. Simülasyon modülünün ardından genetik programlama modülü geliştirilmiştir. Bu modül, simülasyon modülünü de kapsayarak modelin genetik operasyonlar ile gelişmesini ve probleme uygun evrimleşmiş çözümü bulmasını sağlamaktadır. Modüllerin oluşturulmasının ardından genetik programlamada kullanılacak öznitelikler belirlenmiştir. Bu öznitelikler, etkileşim ve tavsiye tabanlı olmak üzere ikiye ayrılmıştır. Etkileşim tabanlı öznitelikler, kullanıcıların birbirleri ile olan geçmiş etkileşimlerine dayalı ve kullanıcıya özel öznitelikleri barındırmaktadır. Tavsiye tabanlı öznitelikler ise kullanıcının başka bir kullanıcı hakkında komşularından tavsiye niteliğiyle aldığı öznitelikleri içermektedir. Belirlenen

öznitelikler ile birlikte genetik programlamada kullanılacak terminal ve fonksiyon kümeleri belirlenmiştir. Son olarak evrimleşen çözümlerin başarısının yeterliliği için uygunluk fonksiyonu belirlenmiştir. Uygunluk fonksiyonu, güven modelinin sisteme uygulanması sonrası elde edilen saldırı sayısı ile güven modeli olmadan yapılan saldırı sayısının oranı şeklinde oluşturulmuştur. Bu oranın düşmesi başarının arttığını ifade etmektedir.

Modelin ve yapıların oluşturulmasının ardından eğitim ve test işlemleri gerçekleştirilmiştir. Bu aşamada iki saldırgan davranışı sistemde uygulanmıştır. Bunlardan ilki basit bir saldırgan olan saf saldırganlardır. Saf saldırganlar, daima saldırıda bulunan ve kötü niyetli tavsiye veren saldırganlardır. İkinci saldırgan davranışı ise ikiyüzlü saldırganlardır. Bu saldırganlar belirli bir olasılık ile saldıran ve kötü niyetli tavsiye veren saldırganlardır. Model öncelikle bireysel olarak hareket eden saf ve ikiyüzlü saldırganlar üzerine eğitilmiş ve test edilmiştir. Testler sonrasında başarılı sonuçlar elde edilmiştir. Ardından bir grup halinde hareket eden işbirlikçi saldırganlar üzerine çalışılmıştır. Saf saldırganlar ilk etkileşimlerinde tespit edilebilmesi kolay saldırganlar olduğu için işbirlikçi avantajından yararlanamamışlardır. İkiyüzlü saldırganlar ise birbirlerine verdikleri itibar arttırıcı tavsiyeler ve arada iyi davranış sergilemeleri sebebiyle işbirlikçi avantajından saf saldırganlara nazaran daha fazla faydalanmışlardır. Bu zor saldırı davranışına karşın ise model başarılı sonuçlar vermiştir.

Eşler arası sistemlerde bir saldırgan sistemde edindiği kötü itibarı sıfırlamak ve varlığını daha uzun süre sürdürebilmek için kimlik değiştirme yöntemini kullanabilmektedir. Tez kapsamında önerilen model kimlik değiştiren saldırgan türlerine karşı eğitilmiş ve test edilmiştir. Belirli aralıklarla kimliklerini değiştiren saldırganlardan saf saldırganlar, hemen tespit edilmelerinin ardından değiştirdikleri kimlikleri sayesinde tekrardan saldırıda bulunarak bu yöntemden fayda sağlamışlardır. İlk etkileşimlerinde hemen tespit edilen ve sistemden dışlanan saf saldırganlar bu şekilde

sisteme tekrar dâhil olabilmişlerdir. İkiyüzlü saldırganlar ise saf saldırganlar kadar kimlik değiştirme yönteminin avantajlarından faydalanamamışlardır. Bunun sebebi verdikleri kötü niyetli tavsiyeler ve iyi davranışları ile edindikleri iyi itibarı bir nebze kaybediyor olmalarıdır. Modelin başarısı yine yüksek olsa da saf saldırganlara karşı elde edilen başarının düşüş oranı ikiyüzlü saldırganlara nazaran fazla olmuştur. Gerçek eşler arası bir sisteme yakın olması açısından karışık saldırgan türlerini içeren ortamlarda da model eğitilmiş ve test edilmiştir. Her bir saldırgan türünden belirli oranlarda bulunan ortamda eğitilen model, yapılan testler sonrasında azımsanmayacak başarı sağlamıştır. Bu da modelin ortamın durumuna göre evrimleştiğini göstermektedir.

Modelin saldırgan türlerine bağlı olarak başarı ölçümünün yanı sıra modeli oluşturan özniteliklerin başarı üzerindeki etkisi de araştırılmıştır. Bunun için etkileşim sonrası oluşturulan memnuniyet ve ağırlık değerleri kullanılmıştır. İki değer de ayrı ayrı çıkartılarak model eğitilmiş ve sonuçlar değerlendirilmiştir. Yapılan eğitim ve testler sonrasında iki değerin de çıkartılması modelin başarısında düşüşe sebep olmuştur. Sonuçlar incelendiğinde memnuniyet değerlerinin ağırlık değerlerine göre başarıyı daha fazla etkilediği görülmüştür.

Modelin testleri sırasında son olarak çapraz eğitim ve test gerçekleştirilmiştir. Bu aşamada iki farklı deney grubu üzerinde çalışılmıştır. Bunlardan ilki, kolay olan saf saldırganlar ile daha zor olan ikiyüzlü saldırganlar arasında gerçekleştirilmiştir. Saf saldırgan türünde eğitilen model hem saf hem de ikiyüzlü saldırganlar üzerinde test edilmiş, ikiyüzlü saldırganlara göre eğitilen model de yine hem saf hem de ikiyüzlü saldırganlar üzerinde test edilmiştir. İkinci deney grubu olarak işbirlikçi saldırılara göre daha kolay olan bireysel ikiyüzlü saldırganlar ile işbirlikçi ikiyüzlü saldırganlar kullanılmıştır. İşbirlikçi saldırgan türünde eğitilen model hem bireysel hem de işbirlikçi saldırganlar üzerinde test edilmiş, işbirlikçi saldırganlara göre eğitilen model de yine hem bireysel hem de işbirlikçi

saldırganlar üzerinde test edilmiştir. Sonuçlar incelendiğinde, kendi türünde eğitilip test edilen modelin yüksek başarı gösterdiği görülmüştür. Fakat kolay bir saldırgan türünde eğitilen model zor bir saldırgan türünde düşük başarı gösterirken, zor bir saldırgan türünde eğitilen model hem kolay hem de zor saldırgan türünde başarılı olmuştur. Sonuçların gösterdiği üzere saldırıların karmaşıklığı ve zorluğu ne kadar fazla olursa modelin direnci de o oranda artmaktadır. Bu da modelin saldırganın zorluğuna göre daha da güçlü çözümler üretebildiğini göstermektedir.

Sonuç olarak yapılan çalışmalar kapsamında eşler arası sistemlerde güven yönetiminin sağlanması adına sabit çözümler yerine ortamın durumuna göre evrimleşebilen ve gelişebilen bir model önerilmiştir. Önerilen modelin başarılı olduğu yapılan deneyler ile gösterilmiştir. Güven yönetimi probleminin çözümü için önerilen genetik programlama yaklaşımının nasıl uygulanması gerektiği araştırılmış ve bu yaklaşım sayesinde başarılı sonuçlar edilmiştir.

KAYNAKÇA

[1] J. F. Kurose and K. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., **2002**.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *SIGCOMM Comput. Commun. Rev.*, vol. 31, pp. 161–172, Aug. **2001**.

[3] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system," in *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, (New York, NY, USA), pp. 310–317, ACM, **2001**.

[4] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati, "Choosing reputable servents in a p2p network," in *Proceedings of the 11th international conference on World Wide Web*, WWW '02, (New York, NY, USA), pp. 376–386, ACM, **2002**.

[5] Clip2, "The gnutella protocol specification v0.4 (document revision 1.2)." http://www.clip2.com/GnutellaProtocol04.pdf, **2001**.

[6] E. Adar and B. A. Huberman, "Free riding on gnutella," *First Monday*, vol. 5, p. 2000, **2000**.

[7] A. A. Selcuk, E. Uzun, and M. R. Pariente, "A reputation-based trust management system for p2p networks," in *Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid*, CCGRID '04, (Washington, DC, USA), pp. 251–258, IEEE Computer Society, **2004**.

[8] R. Zhou, K. Hwang, and M. Cai, "Gossiptrust for fast reputation aggregation in peer-to-peer networks," *IEEE Trans. on Knowl. and Data Eng.*, vol. 20, pp. 1282–1295, Sept. **2008**.

[9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, “The eigentrust algorithm for reputation management in p2p networks,” in *Proceedings of the 12th international conference on World Wide Web*, WWW ’03, (New York, NY, USA), pp. 640–651, ACM, **2003**.

[10] L. Xiong and L. Liu, “Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities,” *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, pp. 843–857, July **2004**.

[11] A. Oram, ed., *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. Sebastopol, CA, USA: O’Reilly & Associates, Inc., **2001**.

[12] J. Liang, R. Kumar, and K. Ross, “Understanding KaZaA,” **2004**.

[13] M. Ripeanu, “Peer-to-peer architecture case study: Gnutella network,” **2001**.

[14] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, “Freenet: a distributed anonymous information storage and retrieval system,” in *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability*, (New York, NY, USA), pp. 46–66, Springer-Verlag New York, Inc., **2001**.

[15] R. E. Tarjan, “Depth-first search and linear graph algorithms.,” *SIAM J. Comput.*, vol. 1, no. 2, pp. 146–160, **1972**.

[16] I. Stoica, R. Morris, D. Liben-nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup protocol for internet applications,” *IEEE/ACM Transactions on Networking*, vol. 11, pp. 17–32, **2003**.

[17] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy, “Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web,” in *In ACM Symposium on Theory of Computing*, pp. 654–663, **1997**.

[18] A. I. T. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Middleware ’01, (London, UK, UK), pp. 329–350, Springer-Verlag, **2001**.

[19] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment,” *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 41–53, **2004**.

[20] M. Deutsch, “The resolution of conflict: Constructive and destructive processes.,” *New Haven: Yale University Press.*, **1973**.

[21] T. Grandison and M. Sloman, “A survey of trust in internet applications,” **2000**.

[22] E. Chang, F. Hussain, and T. Dillon, *Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence*. John Wiley & Sons, **2005**.

[23] A. Abdul-Rahman and S. Hailes, “Supporting trust in virtual communities,” pp. 4–7, **2000**.

[24] J. Sabater and C. Sierra, “Regret: A reputation model for gregarious societies,” pp. 61–69, **2001**.

[25] L. Mui, M. Mohtashemi, and A. Halberstadt, “A computational model of trust and reputation for e-businesses,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS’02)-Volume 7 - Volume 7*, HICSS ’02, (Washington, DC, USA), pp. 188–, IEEE Computer Society, **2002**.

[26] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, “Reputation systems,” *Commun. ACM*, vol. 43, pp. 45–48, Dec. **2000**.

[27] A. Abdul-Rahman and S. Hailes, “A distributed trust model,” in *Proceedings of the 1997 workshop on New security paradigms*, NSPW ’97, (New York, NY, USA), pp. 48–60, ACM, **1997**.

[28] A. Abdul-Rahman and S. Hailes, “Supporting trust in virtual communities,” in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6 - Volume 6*, HICSS ’00, (Washington, DC, USA), pp. 6007–, IEEE Computer Society, **2000**.

[29] E. Damiani, D. C. D. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, “A reputation-based approach for choosing reliable resources in peer-to-peer networks,” in *In Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 207–216, ACM Press, **2002**.

[30] N. Padhy, P. Mishra, and R. Panigrahi, “The survey of data mining applications and feature scope,” *CoRR*, vol. abs/1211.5723, **2012**.

[31] S. R. Michalski, G. J. Carbonell, and M. T. Mitchell, eds., *Machine learning an artificial intelligence approach volume II*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., **1986**.

[32] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2nd ed., **2010**.

[33] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, **1992**.

[34] M. Ruse, “Charles darwin’s theory of evolution: An analysis,” *Journal of the History of Biology*, vol. 8, no. 2, pp. 219–241, **1975**.

[35] M. L. Wong and K. S. Leung, *Data Mining Using Grammar-Based Genetic Programming and Applications*. Norwell, MA, USA: Kluwer Academic Publishers, **2000**.

[36] A. Takac, “Cellular genetic programming algorithm applied to classification task,” *Neural Network World*, vol. 14, pp. 435–452, **2004**.

[37] L. Altenberg, “The evolution of evolvability in genetic programming,” *Advances in genetic programming*, vol. 3, pp. 47–74, **1994**.

[38] X. Liu, G. Tredan, and A. Datta, “A generic trust framework for large-scale open systems using machine learning,” *CoRR*, vol. abs/1103.0086, **2011**.

[39] K. Fukunaga, *Introduction to statistical pattern recognition (2nd ed.)*. San Diego, CA, USA: Academic Press Professional, Inc., **1990**.

[40] G. J. Mclachlan, *Discriminant Analysis and Statistical Pattern Recognition (Wiley Series in Probability and Statistics)*. Wiley-Interscience, Aug. **2004**.

[41] J. R. Quinlan, “Induction of decision trees,” *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, **1986**.

[42] R. Beverly and M. Afergan, “Machine learning for efficient neighbor selection in unstructured p2p networks,” in *Proceedings of the 2Nd USENIX Workshop on Tackling Computer Systems Problems with Machine Learning Techniques*, SYSML’07, (Berkeley, CA, USA), pp. 1:1–1:6, USENIX Association, **2007**.

[43] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Min. Knowl. Discov.*, vol. 2, pp. 121–167, June **1998**.

[44] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., **1995**.

[45] W. Song and V. V. Phoha, “Neural network-based reputation model in a distributed system.,” in *CEC*, pp. 321–324, IEEE Computer Society, **2004**.

[46] R. Rojas, *Neural Networks - A Systematic Introduction*. Berlin: Springer-Verlag, **1996**.

[47] G. Zacharia, “Trust management through reputation mechanisms,” *Applied Artificial Intelligence*, vol. 14, pp. 881–907, **2000**.

[48] R. Luling, B. Monien, and F. Ramme, “A study on dynamic load balancing algorithms,” in *In Proceedings Of The 3rd IEEE SPDP*, pp. 686–689, **1991**.

[49] W. Song, V. V. Phoha, and X. Xu, “An adaptive recommendation trust model in multiagent system.,” in *IAT*, pp. 462–465, IEEE Computer Society, **2004**.

[50] C. Selvaraj and S. Anand, “Peer profile based trust model for p2p systems using genetic algorithm.,” *Peer-to-Peer Networking and Applications*, vol. 5, no. 1, pp. 92–103, **2012**.

[51] D. E. Denning, “An intrusion-detection model,” *IEEE Trans. Softw. Eng.*, vol. 13, pp. 222–232, Feb. **1987**.

[52] M. Crosbie and E. H. Spafford, “Applying genetic programming to intrusion detection,” in *Working Notes for the AAAI Symposium on Genetic Programming* (E. V. Siegel and J. R. Koza, eds.), (MIT, Cambridge, MA, USA), pp. 1–8, AAAI, 10–12 November **1995**.

[53] A. Abraham and C. Grosan, “Evolving intrusion detection systems,” in *Genetic Systems Programming: Theory and Experiences* (N. Nedjah, A. Abraham, and L. de Macedo Mourelle, eds.), vol. 13 of *Studies in Computational Intelligence*, pp. 57–80, Germany: Springer, **2006**.

[54] D. Wilson and D. Kaur, “Knowledge extraction from kdd’99 intrusion data using grammatical evolution,” *WSEAS Transactions on Information Science and Applications*, vol. 4, pp. 237–244, February **2007**.

[55] S. Sen and J. A. Clark, “A grammatical evolution approach to intrusion detection on mobile ad hoc networks,” in *Proceedings of the Second ACM Conference on Wireless Network Security*, pp. 95–102, ACM, **2009**.

[56] S. Sen and J. Clark, “Evolutionary computation techniques for intrusion detection in mobile ad hoc networks,” *Computer Networks*, vol. 55, pp. 3441–3457, oct **2011**.

[57] H. Chen, J. A. Clark, J. E. Tapiador, S. A. Shaikh, H. Chivers, and P. Nobles, “A multi-objective optimisation approach to ids sensor placement,” in *CISIS*, pp. 101–108, **2009**.

[58] S. Sen and Z. Dogmus, “Feature selection for detection of ad hoc flooding attacks.,” in *ACITY (1)* (N. Meghanathan, D. Nagamalai, and N. Chaki, eds.), vol. 176 of *Advances in Intelligent Systems and Computing*, pp. 507–513, Springer, **2012**.

[59] W. Banzhaf, *Genetic programming : an introduction on the automatic evolution of computer programs and its applications*. San Francisco, Calif.; Heidelburg: Morgan Kaufmann Publishers ; Dpunkt-verlag, **1998**.

[60] A. B. Can and B. Bhargava, “Sort: A self-organizing trust model for peer-to-peer systems,” *IEEE Trans. Dependable Sec. Comput.*, vol. 10, no. 1, pp. 14–27, **2013**.

[61] S. Saroiu, K. P. Gummadi, and S. D. Gribble, “A Measurement Study of Peer-to-Peer File Sharing Systems,” in *Multimedia Computing and Networking (MMCN)*, January **2002**.

[62] M. Ripeanu, I. Foster, and A. Iamnitchi, “Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design,” *IEEE Internet Computing Journal*, vol. 6, Jan./Feb. **2002**.

[63] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, “An analysis of internet content delivery systems,” **2002**.

[64] A. B. Can, *Trust and Anonymity in Peer-to-peer Systems*. PhD thesis, West Lafayette, IN, USA, **2007**.

[65] “Ecj 21: A java-based evolutionary computation and genetic programming research system.” http://www.cs.umd.edu/projects/plus/ec/ecj/, **2013**.

[66] M. A. Hall, *Correlation-based Feature Selection for Machine Learning*. PhD thesis, **1999**.

[67] N. L. Cramer, “A representation for the adaptive generation of simple sequential programs.,” in *ICGA* (J. J. Grefenstette, ed.), pp. 183–187, Lawrence Erlbaum Associates, **1985**.

[68] M. Punceva and K. Aberer, “Trust-aware delivery of composite goods.,” in *DBISP2P* (K. Aberer, V. Kalogeraki, and M. Koubarakis, eds.), vol. 2944 of *Lecture Notes in Computer Science*, pp. 219–231, Springer, **2003**.

[69] S. Xiao and I. Benbasat, “The formation of trust and distrust in recommendation agents in repeated interactions: A process-tracing analysis,” in *Proceedings of the 5th International Conference on Electronic Commerce*, ICEC ’03, (New York, NY, USA), pp. 287–293, ACM, **2003**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı	:	Uğur Eray TAHTA
Doğum Yeri	:	Ankara
Medeni Hali	:	Bekar
E-Posta	:	eraytahta@gmail.com
Adresi	:	Gazi Mah. Yenimahalle/Ankara

Eğitim

Lise	:	Turhal Anadolu Lisesi (2001-2005)
Lisans	:	Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü (2005-2010)

Yabancı Dil ve Düzeyi

İngilizce - ileri

İş Deneyimi

2010-... ASELSAN A.Ş. - Yazılım Mühendisi

Deneyim Alanları

C, C++, C#, Assembly, Java, UML, SQL, Ada95, Office araçları, Doors, Clearcase, Enterprise Architect, Rational Rose, Windows OS, Linux (Ubuntu/Kubuntu, Fedora), MS-DOS, MS Visual Studio 6.0/2005/2008/2010, Eclipse, NetBeans

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu İle Katıldığı Toplantılar

Tebliğ : Evolving a Trust Model for Peer-To-Peer Networks Using Genetic Programming

Yer : Evo2014 - EvoStar, İspanya - 2014