VERİ MADENCİLİĞİ İLE BİLGİSAYAR AĞLARINDA YENİ BİR SALDIRI TESPİT ALGORİTMASI

Nurullah Celal USLU

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MAYIS 2009
ANKARA

Nurullah Celal USLU tarafından hazırlanan VERİ MADENCİLİĞİ İLE BİLGİSAYAR AĞLARINDA YENİ BİR SALDIRI TESPİT ALGORİTMASI adlı bu tezin Yüksek Lisans Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. M. Ali AKCAYOL …….
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından oy çokluğu ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Prof. Dr. Hadi GÖKÇEN …….
Bilgisayar Mühendisliği, Gazi Üniversitesi

Prof. Dr. Şeref SAĞIROĞLU …….
Endüstri Mühendisliği, Gazi Üniversitesi

Doç. Dr. M. Ali AKCAYOL …….
Bilgisayar Mühendisliği, Gazi Üniversitesi

Tarih: 18/05/2009

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nail ÜNSAL …….

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Nurullah Celal USLU

VERİ MADENCİLİĞİ İLE BİLGİSAYAR AĞLARINDA YENİ BİR SALDIRI TESPİT ALGORİTMASI

(Yüksek Lisans Tezi)

Nurullah Celal USLU

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Mayıs 2009

ÖZET

Bu çalışmada veri madenciliği tekniklerinden biri olan karar ağaçlarıyla bilgisayar ağlarında yeni bir saldırı tespit algoritması geliştirilmiştir. Standart ID3 ve önerilen algoritmaya göre oluşturulmuş karar ağaçları KDD Cup'99 verisiyle test edilmiş ve her iki algoritmanın saldırı tespit başarımlarının raporlanması gerçeklenmiştir. Deneysel sonuçlardan da görüldüğü üzere yeni önerilen algoritmanın şu an uygulamada kullanılan ID3 algoritmasından daha iyi bir başarım gösterdiği görülmüştür.

Bilim Kodu : **902.1.014**
Anahtar Kelimeler : **Saldırı tespit sistemleri, veri madenciliği, ID3, saldırı tespit algoritması, saldırı veri kümeleri, C4.5**
Sayfa Adedi : **87**
Tez Yöneticisi : **Doç. Dr. M. Ali AKCAYOL**

A NEW INTRUSION DETECTION ALGORITHM
IN COMPUTER NETWORKS WITH DATA MINING
(M.Sc. Thesis)

Nurullah Celal USLU

GAZI UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY
May 2009

ABSTRACT

In this study, a new intrusion detection algorithm in computer networks has been developed with decision trees which is one of the data mining techniques. Decision trees which are generated according to standard ID3 and proposed algorithm have been tested with KDD Cup'99 test data sets and both of these algorithms' performance of intrusion detection has been reported. As the experimental studies show the proposed algorithm has become more successful than the standard ID3 algorithm.

Science Code : 902.1.014
Key Words : Intrusion detection systems, data mining, ID3, intrusion detection algorithm, intrusion data sets, C4.5
Page Number: 87
Adviser : Assoc. Prof. Dr. M. Ali AKCAYOL

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren Danışman Hocam Sayın Doç. Dr. M. Ali AKCAYOL'a, Yüksek Lisans eğitimim süresince yardımlarını esirgemeyen hocalarım Sayın Prof. Dr. Şeref SAĞIROĞLU ve Sayın Yrd. Doç. Dr. Hasan Şakir BİLGE'ye, çalışmam sırasında desteklerini esirgemeyen arkadaşlarım Mehmet ÖZKAR, Samet KARAKAYNAK ve Mehmet ŞİMŞEK'e ayrıca maddi ve manevi her türlü destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli aileme teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
ATM	Automatic Teller Machine
CERT/CC	Computer Emergency Response Team Coordination Center
COM	Component Object Model
CRISP-DM	The Cross- Industry Standard Process for Data Mining
CRM	Customer Relationship Management
CRC	Cyclic Redundancy Check
DMQL	Data Mining Query Language
FBI	Federal Bureau of Investigation
ICMP	Internet Control Message Protocol
IDES	Intrusion Detection Expertsystem
IDEVAL	Intrusion Detection Evaluation Dataset
IIS	Internet Information Services
ISS	Internet Security Systems
KDD	Knowledge discovery and Data Mining
MAC	Media Access Control
MIT	Massachusetts Institute of Technology
NIDES	Next-Generation İntrusion Detection Systems
POS	Point of Sales
SMS	Short Messaging Service
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
STS	Saldırı Tespit Sistemleri

1. GİRİŞ

Modern iletişim teknolojileri sayesinde kullanıcılara sunulan hizmetlerin sayısı artmıştır. Özellikle ağ teknolojilerinde meydana gelen gelişmeler ağ üzerinden sunulan hizmetlerin sayısını ve çeşitliliğini artırmıştır. Ağ üzerinden sunulan hizmetlerde çeşitlilik beraberinde karmaşıklığı getirmektedir [1].

Günümüzde ağ teknolojileri ve dağıtık sistemler teknolojileri günlük yaşamın her alanında artan bir şekilde kendini göstermektedir. Sağlık kayıtları, banka hesapları vb. kritik bilgilerin işlenmesi dağıtık sistemlerle idare edilmektedir. Bu yüzden bu sistemlerin bulunurluğu(availabilty), doğruluğu ve eksiksizliği son derece önemlidir. Dağıtık sistemlerin karmaşıklığının artmasıyla beraber bu sistemlerin doğruluğunu ve eksiksizliğini denetlemek ve kesintisiz bir şekilde hizmet vermesini sağlamak güçleşmektedir. Bütün bunlara ek olarak dağıtık sistemler tarafından işlenen bilgilerin güvenliği de karmaşıklığı arttıran bir diğer unsurdur. Doğruluğun ve eksiksizliğin denetiminin yapılamadığı durumlarda bilgi güvenliğinin ana eksenleri olan gizlilik, bütünlük ve bulunurluluğa ilişkin gereksinimlerin tam olarak karşılanamaması durumunda kurumlar ciddi tehlikelerle karşı karşıya kalabilirler [1].

Bilgi çağını yaşadığımız şu günlerde gerek hız ve verimlilik artışı, gerekse kolay erişilebilmesi nedeniyle birçok bilgi elektronik ortamlarda saklanmaktadır. Ancak, elektronik ortamlarda saklanan kişisel ya da kurumsal açıdan önemli olan bilginin başkalarının eline geçmesi durumunda çok ciddi maddi ve manevi zararlara yol açabilir. Bu nedenle elektronik ortamlarda saklanan bilgiler için bu riskin minimuma indirilebilmesi için bu bilgilerin güvenliğinin sağlanması gerekmektedir. Elektronik ortamların yaygınlaşarak kullanılmaya başlamasıyla birlikte zaten önemli olan “bilgi ve bilgisayar güvenliği” kavramının önemi gün geçtikçe artmaktadır. [2].

İnternetin ilk yıllarından itibaren fazla önem verilmeyen güvenlik konusu, özellikle 1988 yılındaki Morris Kurdu faciasından sonra “bilgi güvenliği”, “bilgisayar güvenliği” kavramları konuşulmaya başlanmıştır. Kurt, binlerce bilgisayara gizlice sızarak bilgisayar sitemlerini çalışmaz hale getirmiştir. Morris Kurdu ile yaşanan bu

faciadan sonra 1990'lı yılların başlarında ilk güvenlik duvarı uygulamalarıyla bilgi ve bilgisayar güvenliği konusunda ilk çalışmalar başlamıştır [1, 3-5].

Saldırı Tespit Sistemleri(STS), saldırılara karşı sistemimizi koruyan, alarm görevi gören yazılım ve donanımlardır. STS'lerin kullanılması, sistemlere yapılan yetkisiz erişimlerin ve kötüye kullanımların tespit edilmesini, bunların yol açabileceği zararların önüne geçilmesini sağlamaktadır. Ayrıca, STS'lerin kullanılması sistemlere ne tür saldırıların yapıldığı ve sistemlerdeki mevcut açıklar hakkında istatistikî bilgiler elde etmemizi, saldırganlar hakkında detaylı bilgi sahibi olmamızı ve bu bilgiler ışığında sistemimizi tekrar yapılandırmamızı sağlamaktadırlar [6].

STS'ler, ilk olarak 1980'de Anderson'un yaptığı çalışmalar sonucunda ortaya çıkmış ve bu çalışmaların ardından yapılan birçok çalışmayla gelişmesini hızla devam ettirmiştir [69]. 1980'lerde yapılan çalışmalarda sistemler ve ağlar kısmi olarak birbirinden bağımsız haldeydi. Bu dönemde yapılan araştırmalar daha çok konak(host) tabanlı bir saldırı tespite odaklanmıştı. STS'lerin kullanacağı veriler işletim sistemi tarafından oluşturulan uygulama ve güvenlik loglarıyla ağ denetleme verisinden oluşmaktadır. 1990'larda STS yapısında ve ortamında bazı değişiklikler oldu. Bunlardan birincisi ağ STS'lerinin konak tabanlı sistemlerle birlikte üretilmesidir. İkincisi ise web ile ilgili olan bir gelişmedir. İnternetin gelişmesine bağlı olarak güvenlik ihtiyaçları da arttı. Bu dönemde güvenlik duvarları için resmi standartlar belirlendi. Bunun dışında 1990'larda konak tabanlı ve ağ tabanlı saldırı tespit sistemlerinin birleştirilmesi de gerçekleştirilen bir diğer gelişmedir [7].

Günümüze kadar STS'ler üzerinde yapılan çalışmalar incelendiğinde veri toplama, etiketleme, depolama, filtreleme ve sınıflandırma yoluyla veri azaltma, davranış modellerlinin belirlenmesi ve sınıflandırılması, kural tabanlı sistemler için kuralların belirlenmesi, raporlama ve sonuç üretme aşamalarında bir takım güçlüklerle karşılaşılmaktadır. Bu güçlüklerin orta çıkmasının en önemli nedeni ise STS'lerin tasarım, uygulama ve test aşamalarında kullanılabilecek veritabanlarının eksikliğinden kaynaklanmaktadır [8]. Uygulamalarda belli bir standardı olan bir veritabanının olması, gerçekleştirilmesi planlanan saldırı tespit sisteminin

başarımının testinin kısa sürede tamamlanması açısından önemlidir. Bununla birlikte, STS tasarımı sırasında belli bir standardı olmayan, objektiflikten uzak farklı veritabanlarının geliştirilmesi ve buna bağlı olarak objektif bir şekilde sistemin başarım oranının değerlendirilememesinden dolayı standart saldırı veri tabanları üzerine de çalışmalar yapılmıştır.

Bugünün dünyasında şirketlerin tamamına yakını varlıklarını devam ettirebilmek için internete bağımlıdırlar. İnternete bağımlılığın sonucu olarak ağ saldırı tespiti sistemlerinin böylesine gelişmiş olması hatta daha da çok gelişeceği bilinen bir gerçektir [7,9]. 1990'ların STS'lerinde birkaç farklı teknik kullanılırken bu tekniklerin sayısı gün geçtikçe artmaktadır. Kullanılan bu tekniklerin avantajları olduğu gibi dezavantajları da bulunmaktadır. Son zamanlarda saldırı tespitinde etkin olarak kullanılan veri madenciliği de bu tekniklerden birisidir. Bu tez kapsamında da veri madenciliğinin sınıflandırma tekniklerinden biri olan karar ağaçları kullanılarak yeni bir saldırı tespit algoritması geliştirilmiştir.

Veri madenciliğinin en önemli ham maddesi olan verinin toplanması işlemi bilgi teknolojilerindeki gelişmenin sonucu olarak bugünün dünyasında çok kolay bir hale gelmiştir. Bilgi teknolojilerinin gün geçtikçe daha yaygın bir şekilde kullanılmasıyla birlikte bilgilerin çoğu elektronik ortamlarda saklanmaya başlanmıştır. Verilerin dijital ortamda saklanmasıyla birlikte, bilgi miktarı her 20 ayda bir iki katına çıktığı günümüzde verileri saklama ve işleme olarak kullanılan veri tabanlarının sayısı ve boyutu da benzer bir şekilde hatta daha yüksek oranda artmaktadır. Ayrıca bilgi teknolojilerinde yaşanan bu gelişmeler, yüksek kaliteye sahip teknolojik ürünlerin ilk fiyatlarına nazaran daha makul fiyatlara inmesini sağlamıştır. Sadece sağlık ve astronomi sektöründe veri tabanının hacminin ulaştığı boyutlar bile çok ciddi rakamlara ulaşmıştır. Bu iki sektörün veri tabanı hacminin bu kadar büyük olduğunu düşündüğümüzde bu veri tabanlarından anlamlı örüntülerin keşfinin insanların veya ilişkisel veri tabanı teknolojilerinin yapabileceği bir iş olmadığı anlaşılmaktadır [10,11].

Veri yığınının fazla bir anlamı yoktur. Önemli olan amacımız doğrultusunda erişebileceğimiz bilgidir. Bilgi, bir amaca yönelik işlenmiş veridir; bir başka deyişle bir soruya yanıt verebilmek için veriden çıkardığımız anlamlı veri olarak tanımlanabilir. Büyük miktardaki verileri işleyebilen teknikleri kullanabilmek günümüzde ihtiyaç haline gelmiştir. Bu ihtiyacın giderilmesi için araştırma kurumları ve üniversiteler bu konuda çeşitli çalışmalar yaparak yeni disiplinlerin ortaya çıkmasını sağlamışlardır. Veri madenciliği(VM) de bu yeni disiplinlerden birisidir. Bu yeni disiplin sayesinde anlamsız olan veri yığınının anlamlı hale gelmesi sağlanabilmektedir. Örneğin eskiden süpermarketlerdeki kasa basit bir toplama makinesinden ibaretken, günümüzde ise kasa yerine kullanılan satış noktası terminalleri sayesinde müşteriye ait hareketlerin bütün detayları saklanabilmektedir. Böylelikle müşterilere ait verilere istenildiği zaman ulaşmak ve bu verileri analiz etmek mümkün hale gelmiştir [11-13].

Süpermarket örneğinde belirtildiği gibi müşterilere ait veriler analiz edilerek her mal için bir sonraki ayın satış tahminleri yapılabilir; müşterilerin satın aldıkları mallara göre gruplandırılması yapılabilir; yeni çıkarılacak ürünün potansiyel müşterileri belirlenebilir ve müşterilerin zaman içindeki hareketleri incelenerek bu müşterilerle ilgili geleceğe yönelik tahminler yapılarak istatistikî veriler elde edilebilir. Binlerce malın ve müşterinin olduğu düşünüldüğünde bu analizler ve tahminlerin elle yapılmayacağı, otomatik olarak yapılmasının gerektiği ortaya çıkmaktadır. İşte burada VM devreye girmektedir. Büyük miktarda veri yığınlarından anlamlı örüntülerin çıkarılması, ihtiyacımız olan bilginin elde edilmesini sağlayan VM, süpermarket örneğindeki işlemlerin de otomatik bir şekilde yapılmasını sağlayan bir disiplindir [13].

Bu tez çalışması 6 bölümden oluşmuştur. Tezin ikinci bölümünde, VM'nin literatürdeki farklı tanımları, tarihi gelişimi, VM süreci, VM'de karşılaşılabilecek önemli sorunlar, ilişkide olduğu disiplinler, uygulama alanları ve VM algoritmaları üzerinde detaylı olarak durulmuştur.

Üçüncü bölümde, STS'lerin tarihsel gelişimi ve bilgi ve bilgisayar güvenliği açısından öneminden bahsedilmiş, bilinen saldırı tipleri ve STS türleri, STS'nin önemi üzerinde durulmuştur.

Dördüncü bölümde, STS veri kümeleri, standart ID3 algoritması ve ID3 algoritmasının temelini oluşturan karar ağaçları, C4.5 algoritması ve bu tez kapsamında geliştirilen saldırı tespit algoritması detaylı bir şekilde anlatılmıştır.

Beşinci bölümde, geliştirilen bu algoritmanın farklı veri miktarlarıyla test edilerek genel başarımı ve her saldırı grubu için başarımı ayrı ayrı raporlanmış ve bu deneysel sonuçlar grafikler ve şekillerle sunulmuştur.

Altıncı ve son bölümde, bu tez çalışması kapsamında karşılaşılan güçlüklerden ve kazanımlardan bahsedilmiştir. En son olarak bu çalışmanın sonuçları ve gelecek çalışmalar için bazı öneriler sunulmuştur.

Tezin sonunda iki adet ek vardır. Ek-1'de dördüncü bölümde geliştirilmiş olan uygulamada kullanılan bazı önemli fonksiyonların ve değişkenlerin açıklamaları sunulmuştur. Ek-2'de ise dördüncü bölümde bahsedilen saldırı tespit algoritmasıyla oluşturulan karar ağacının belli bir bölümünün resmi sunulmuştur.

2. VERİ MADENCİLİĞİ

Bilgi teknolojilerinde yaşanan gelişmeler sonucunda bilgisayar sistemleri hem ucuzlamış hem de performansları artmıştır. Artık bilgisayarlar daha büyük miktardaki veriyi daha kısa zamanda işleyebilmektedir. Diğer taraftan bilgisayar ağlarındaki ilerlemeler ile bir bilgisayardaki veriye başka bir bilgisayardan daha hızlı erişilmektedir. Bütün bunların yanında bilgisayarların ucuzlaması da insanların sayısal teknolojiyi daha yaygın olarak kullanmasını sağlamıştır [13]. Böylelikle veri doğrudan toplanmakta ve elektronik ortamlarda saklanmaktadır. Verinin bu kadar kolay ve hızlı bir şekilde elde edilmesi sonucunda dijital ortamlarda toplanan verinin miktarı gün geçtikçe artmaktadır.

Ticari, tıp, askeri, iletişim ve astronomi gibi birçok alanda veri hacminin yaklaşık yirmi ayda bir iki katına çıktığı tahmin edilmektedir. Veriler o kadar hızlı bir şekilde toplanıyor ki bunun en açık örneğini NASA kurumunda görmekteyiz. NASA'nın kullandığı uyduların sadece birinden bir günde alınan verilerin terabayt'lar düzeyinde olduğu bilinmektedir [14]. Zaman geçtikçe hızla artan bu verilerin saklanması ve işlenmesi problem haline gelmiştir. Bu probleme veri tabanları ve dosya sistemlerindeki gelişmelerle çözüm aransa da tam anlamıyla bir çözüm bulunamamıştır. Donanımların ucuzlamasına paralel olarak veri tabanlarının sayısında ve hacimlerinde de ciddi bir artış olmaktadır. Bu artışın olması bizim sorunumuzu çözmektense anlamlı olmayan veri yığınının artışına sebep olmaktadır. Gerek veri tabanı teknolojisindeki gelişmeler gerekse donanımın ucuzlaması veri yığınlarının artmasına sebep olmuştur. Bu durumda veri yığınından anlamlı örüntülerin çıkarılması, onların nasıl anlamlı hale getirileceği sorunu orta çıkmıştır [15,16].

Veri tabanlarında tutulan bu veri yığınlarından anlamlı bilgiler elde etmek ve karar destek aşamasında faydalanmak, herhangi bir araç kullanmadan, manüel olarak bunu gerçekleştirmek çok güçtür. Veri tabanlarında gerçek hayatta işe yaramayan verilerin tutulması insanlar için ek bir maliyetten başka bir şey değildir. İşte bu noktada karşımıza çözüm olarak, üniversitelerin ve araştırma kurumlarının bu problemi

çözmek için ortaya çıkardıkları yeni bir disiplin olan VM çıkmaktadır [15]. VM, belli bir amaç için toplanmış bir veri yığınından veriler arasında gizli kalmış anlamlı bilgilerin ortaya çıkarılmasıdır. Ayrıca, geleceğin geçmişten çok farklı olmayacağı düşünülerek geleceğe yönelik tahminlerde bulunarak kararlar almamızda bize fikir verir. VM doğası gereği istatistik, veri tabanları, makine öğrenmesi, bilgi toplama, görselleştirme, paralel ve dağıtık hesaplama gibi birçok disiplinle ilişkilidir. [17].

Aşağıda VM için literatürdeki bazı tanımlamalar verilmiştir.

- VM'nin konusu, büyük veri yığınlarından statiksel kural ya da örüntü biçimindeki bu tür bilgilerin çıkarılmasıdır [10].
- VM, eldeki verilerden üstü kapalı, çok net olmayan, önceden bilinmeyen ancak potansiyel olarak kullanışlı bilginin çıkarılmasıdır. Bu da; kümeleme, veri özetleme, değişikliklerin analizi, sapmaların tespiti gibi belirli sayıda teknik yaklaşımları içerir [18].
- VM, verilerin içerisindeki desenlerin, ilişkilerin, değişimlerin, düzensizliklerin, kuralların ve istatistiksel olarak önemli olan yapıların yarı otomatik olarak keşfedilmesidir [19].
- VM büyük miktarda veri içinden gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanarak aranmasıdır [11].
- VM, büyük hacimli veri içerisinden; anlamlı, gizli kalmış ve kuruluşun karar destek sistemi için potansiyel olarak faydalı olabilecek bilgilerin çıkarıldığı ve geri planında istatistik, yapay zekâ ve veritabanlarının bulunduğu veri analiz tekniğidir [7].
- VM, istatistik ve matematik tekniklerle birlikte ilişki tanıma teknolojilerini kullanarak, depolama ortamlarında saklanmış bulunan veri yığınlarının elenmesi ile anlamlı yeni ilişki ve eğilimlerin keşfedilmesi sürecidir [20].
- VM, önceden bilinmeyen, geçerli ve uygulanabilir bilginin veri yığınlarından dinamik bir süreç ile elde edilmesi olarak tanımlanabilir. Bu süreçte kümeleme, veri özetleme sınıflama kurallarının öğrenilmesi, bağımlılık ağlarının bulunması

ağlarının bulunması, değişkenlik analizi ve anomali tespiti gibi farklı birçok teknik kullanılmaktadır [20].

- VM, kendi başına bir çözüm değil çözüme ulaşmak için verilecek karar sürecini destekleyen, problemi çözmek için gerekli bilgileri sağlamaya yarayan bir araçtır [21].
- VM, veriden anlamlı ilişkiler ve örüntüler (patterns) çıkarma sürecine, "veri madenciliği", "bilgi çıkarımı", "bilgi keşfi", "veri arkeolojisi" ve "veri şablon işleme" gibi isimler verilmektedir. Veri madenciliği tanımını daha çok istatistikçiler, veri analizcileri ve yönetim bilişim sistemleri kullanıcıları kullanmaktadır. İlk olarak 1989 yılında bir atölye çalışmasında, veri işleme sürecinde bilginin son ürün olduğunu vurgulamak için "veri tabanlarında bilgi keşfi" tanımlaması kullanılmıştır [22].
- VM, büyük miktardaki verinin, bu veriden anlamlı örüntü ve kurallar çıkarılması amacıyla analiz edilmesidir [23].
- VM, veri tabanlarında zengin bilgiye sahip olan pek çok organizasyon, bu bilgiyi yönetmenin çok zor olması sebebiyle, bilgisayarları kullanmaktadır. Bilgisayarların kullanılarak veriler içerisinden anlamlı bilgilerin çıkarılması, veri madenciliği olarak tanımlanmıştır [24].
- VM, büyük miktarda veri içinden, gelecekle ilgili tahmin yapmamızı sağlayacak bağıntı ve kuralların bilgisayar programları kullanılarak aranmasıdır. Veri analizi yapılarak, bir mal için bir sonraki ayın satış tahminleri yapılabilir, müşteriler satın aldıkları mallara bağlı olarak gruplandırılabilir, yeni bir ürün için potansiyel müşteriler belirlenebilir, müşterilerin zaman içindeki hareketleri incelenerek onların davranışları ile ilgili tahminler yapılabilir. Binlerce malın ve müşterinin olabileceği düşünülürse bu analizin gözle ve elle yapılamayacağı, otomatik olarak yapılmasının gerektiği ortaya çıkar ve veri madenciliği bu noktada devreye girer [13].
- VM, büyük miktarda ve oldukça hızlı toplanan verilerin, çeşitli analizler sonucunda anlamlı bilgilere dönüştürülmesi noktasında devreye giren bir süreçtir. VM tanımları incelendiğinde; bu tanımlarda ortak olan unsurlardan ilki çok fazla

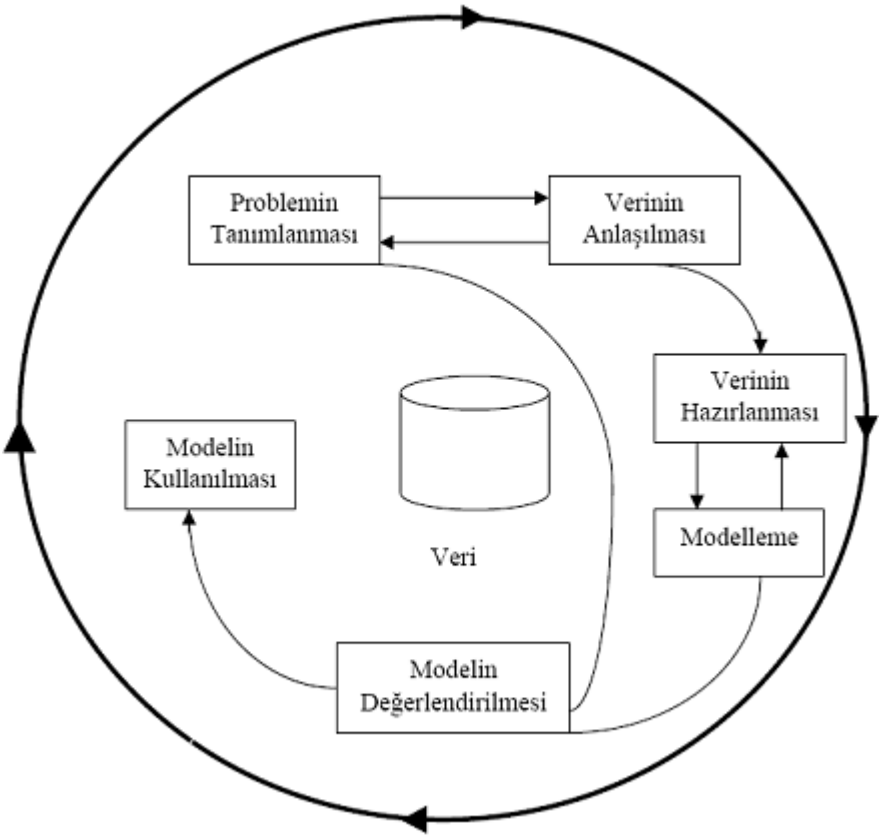
miktarda verinin veri ambarında tutulması, ikincisi ise bu verilerden anlamlı bilgiler elde edilmesidir [25].

- VM, veri yığınları içinden kuruluş yöneticileri için en gerekli olan verilerin seçilmesi, düzenlenmesi ve modellenmesi süreçlerini içermektedir. Veri madenciliği, karar vericilerin kullanabileceği yeni bilgiler oluşturabilmek için yapay zekâ gibi ileri teknoloji içeren yöntemler kullanmaktadır [26].
- VM, arşivlenen bilgiler üzerinde yapılan analizlerle önceden bilinmeyen, değerli ve anlaşılabilir sonuçlar çıkarma süreci olarak tanımlanmıştır. Elde edilen sonuçlar tahmin yürütme, sınıflandırma veya kayıtlar arasındaki benzerliklerin bulunması amacıyla değerlendirilmekte ve bu özellikler karar destek sistemlerinde kullanılmaktadır [26].

2.1. Veri Madenciliği Süreci

VM gizli bilgilerin keşfedilerek ortaya çıkarılması sürecidir. Bu sürecin bazı aşamaları vardır. Her aşama bir sonraki aşamayla ilgilidir, bir önceki aşamanın sonucu bir sonraki aşamanın girdisidir.

VM süreci için farklı görüşler ve uygulamalar söz konusudur. Bu yüzden veri madenciliği süreci için belli bir standart yoktur. Bu standart süreci The Cross-Industry Standard Process for Data Mining (CRISP-DM) konsorsiyumu belirlemiştir. Oluşturulan bu standart süreç CRISP-DM süreci olarak da bilinmekte olup toplam 6 aşamadan oluşmaktadır. Bu süreç Şekil 2.1'de sunulmuştur [26,27].



Şekil 2.1. Veri madenciliği süreci [26]

VM standart süreci aşağıdaki aşamalardan oluşmaktadır:

2.1.1. Problemin tanımlanması

VM sürecinin en önemli aşamasıdır. Veri madenciliği çalışmalarında başarılı olmanın ilk şartı, bu çalışmanın ne tür işletme amacına yönelik olduğunun açık bir şekilde belirlenmesidir [28].

2.1.2. Verinin hazırlanması

VM sürecinin en önemli aşamalarından biri olan verinin hazırlanması aşamasında analist zamanının nerdeyse %50 - %75'ini harcar. Bu aşamada bilgi sistemleri üzerinde üretilen sayısal bilginin iyi analiz edilmesi, veriler ile mevcut iş problemi birbiriyle ilişkili olmalıdır [29].

VM süreci işlemeye devam ederken ortaya çıkabilecek bazı sorunlar, bu sürecin ilk aşaması olan verinin hazırlanması aşamasına dönmesine sebep olabilmektedir. Bu yüzden bu aşamanın önemi çok büyüktür. Verinin hazırlanması; verinin toplanması, verinin birleştirilmesi ve temizlenmesi ile verinin dönüştürülmesi aşamalarından oluşmaktadır [30].

Verinin toplanması: Farklı kaynaklardaki verilerin bir kaynakta birleştirilmesi verinin toplanması olarak ifade edilmektedir [17]. Veri madenciliği sürecinde verinin hazırlanması aşamasının ilk adımı verinin toplanmasıdır. Bu aşamada elde var olan ve toplanması gereken verilerin belirlenmesi ve önemli kararların verilmesi gerekmektedir. Kullanılacak verinin belirlenmesinde veri analizi yapacak kişi veri madenciliğinin hedeflerini ve işletme amaçlarını göz önüne almalıdır [31]. Ayrıca, verinin hangi kaynaklardan toplanacağının önceden belirlenmesi ve bu kaynakların güvenilir olması ileriki aşamalarda meydana gelebilecek problemlerle karşılaşılma riskini azaltmaktadır [26].

Verinin birleştirilmesi ve temizlenmesi: Veri madenciliğinde kullanılacak verilerin farklı kaynaklardan toplanması sonucunda veri uyumsuzluğunun olması tabidir. Veri madenciliğinde birlikte kullanılabilmeleri için bu verilerin birleştirilmesi ve temizlenmesi gerekmektedir [30].

Çizelge 2.1'de bir veri tabanında müşterilerin cinsiyetlerine göre kodlanma biçimleri gösterilmektedir.

Çizelge 2.1. Veri tabanlarında olası kodlama biçimleri [26]

ERKEK	**KADIN**
0	1
1	0
1	2
"ERKEK"	"KADIN"
"erkek"	"kadın"

Çizelge 2.1'de gösterilen kodlama biçimlerindeki farklılık açık bir şekilde görülmektedir. Verilerin farklı kaynaklardan toplanması halinde bu tip sorunlarla karşılaşılabilir. Bu durumda veri analizini yapan kimse gerekli dönüşümleri yapmalıdır. Bunun dışında meydana gelebilecek bir diğer sorun ise eksik veya kayıp bilgilerdir. Örneğin bir veri tabanında kayıtlı bulunan bazı kişilerin medeni halleri belirliyken, bazı kayıtlarda bu bilgiler eksik ya da hiç girilmemiş olabilir. Bu durum "kayıp veriler" olarak nitelendirilmektedir. Bunun dışında bazı kayıtlarda bazı bilgiler örneğin bir kişinin yaşına ait veriler olası değerlerin dışında girilmiş veya yanlış girilmiş olabilir. Bu tür bilgilere de gürültü(noise) adı verilmektedir. Buna ek olarak, bir veri tabanında kayıtlı kişilerin hem yaşları hem de doğum tarihleri tutuluyorsa, burada veri tekrarı vardır. Bu durumda bu değişkenlerin birleştirilerek ya doğum tarihi ya da yaş değişkeni kullanılarak tek bir değişkende tutulması sağlanmalı ve veri tekrarının olması engellenmelidir. Eksik veriler ya zaman içinde ya da tahmin yöntemiyle tamamlanarak eksiklikler giderilebilir [31-33].

Veriyi dönüştürme: Dönüştürme aşamasında kullanılacak modelle veri tabanında tutulan bazı kolonlardaki bilgilerin formatlarının uygun bir şekle dönüştürülmesi aşamasıdır. Örneğin 0/1 şeklinde tutulan bir verinin modele uygun olarak Evet/Hayır şekline dönüştürülmesi sonuç açısından daha faydalı olabilir [12].

2.1.3. Modelin kurulması

Veri madenciliğinde iyi kurulmuş bir model, veri analizi sonucunda elde edilecek sonuçların kalitesini de etkileyecektir. Model kurma süreci, veri analizi için hazır olan verilerin kullanıcıya sunulmasını sağlar. Bu aşamada modelin doğru kurulması çok önemlidir. Eğer model doğru kurulmazsa, veri yığını içerisinde bulunabilecek ilişkiler doğru bir şekilde tespit edilemez ve aranılan doğru örüntü bulunamaz. Buna bağlı olarak kurulan modelden başarılı bir sonuç elde edilme olasılığı azalır [34].

2.1.4. Modelin kullanılması

Veri madenciliği sürecinin son aşamasında modelin kullanılması sağlanır. Bu aşama ile kurulan model belli bir süre izlenir ve modelin yorumlanması ve doğrulanması bu aşamada yapılır [12].

2.2. Veri Madenciliğinde Karşılaşılabilecek Önemli Sorunlar

Veri madenciliğinin yapılabilmesi için işlenmemiş veri yığınlarından oluşan büyük veri tabanlarına ihtiyaç duyulmaktadır. Kaynakları farklı olan bu veri yığınlarının işlenmesi sırasında bazı sorunların çıkması muhtemeldir [36]. Aşağıda veri madenciliğinde ortaya çıkabilecek bazı sorunlardan bahsedilmiştir.

Veri Tabanlarının Boyutları (dimensions of databases): Veri tabanlarının boyutları veri madenciliği işlemlerinde karşılaşılan en büyük problemlerden biridir ve bu problemin çözülmesi için veri tabanlarının boyutlarının küçültülmesi yoluna gidilmiştir. Bir veri tabanının boyutu 2 yolla küçültülebilir:

- Veri alanında örnekleme: Rasgele bazı kayıtlar seçilir ve veri madenciliğinin sonraki aşamalarında kullanılır.
- Özellik alanında örnekleme: Her kayıtın sadece bazı özellikleri seçilir [26].

Dinamik Veri Yapısı (dynamic data structure): Veri tabanları daima dinamik olma eğiliminde oldukları için daima değişiklik gösterebilmektedirler.

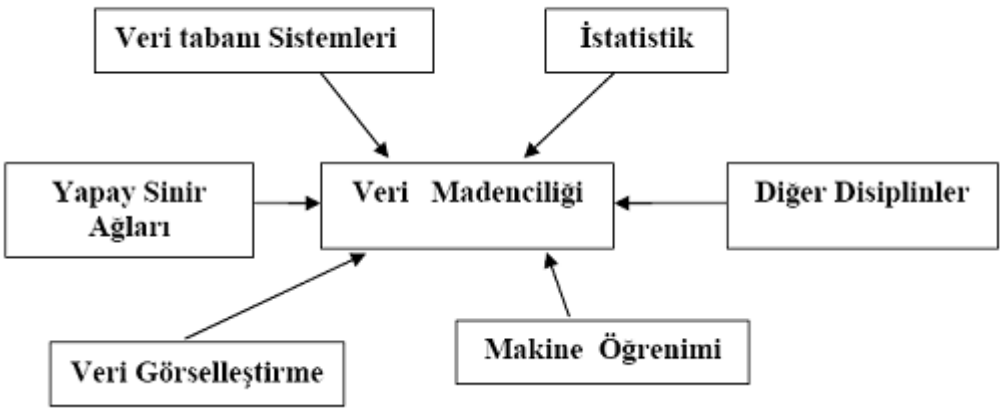
Eksik Veya Kesin Olmayan Veri (missing or uncertain data): Veri tabanlarında toplanan bazı bilgiler eksik olabilir. Veri madenciliğinde kullanılan veri tabanlarının bu eksikliklerinden dolayı bazı problemler ortaya çıkmaktadır. Çünkü veri madenciliğinin amacı toplanan veri yığınlarından anlamlı bilgilerin ortaya çıkarılmasını sağlamaktır. Ancak, eksik değerlere sahip bilgiler üzerinden bu çıkarımın yapılması çok zordur. Bu yüzden eksik veriler veri madenciliği sürecinin başarılı bir şekilde sonuçlanması için oldukça önemlidir [35].

Gürültü (noise): Bazı niteliklerin değerlerindeki veya kayıtların sınıf bilgilerindeki hatalar "gürültü" olarak adlandırılır. Herhangi bir veri toplama tekniğinin gürültüden tamamen arınmış olması pek mümkün olmasa da veri madenciliği işleminin sonucunda ortaya çıkan bilginin doğruluğu için sınıf bilgilerindeki gürültünün arındırılmış olması gerekmektedir [33].

Eksik Değerler (missing values): Veri tabanında tutulan bir niteliğin değeri bilinmiyor olabilir ya da yanlışlıkla girilmemiş olabilir. Veri madenciliği işlemlerinden anlamlı örüntülerin çıkarılması için bu eksik değerlerin yerine ya varsayılan değerlerin koyulması ya da eksik değerlerin bulunduğu kayıtların silinmesi gerekir [35].

2.3. Veri Madenciliği ve İlişkide Olduğu Disiplinler

Veri madenciliği, Şekil 2.2'de görüldüğü gibi veri tabanı sistemleri, istatistik, yapay sinir ağları, makine öğrenimi, veri görselleştirme gibi konuları kapsayan bir disiplindir [34].



Şekil 2.2. Veri madenciliği ve ilişkide olduğu disiplinler [26]

Veri Tabanı Sistemleri: Veri tabanı sistemleri, bilgisayar ortamında büyük boyutlu veri kümelerinin yönetiminde yaygın bir biçimde kullanılan bir araçtır. Bilgi teknolojilerindeki gelişmelerle birlikte veri tabanı sistemleri teknolojilerinde de ciddi

gelişmeler olmuştur. Veri tabanlarının kullanımının en önemli amaçlarından biri de veri tekrarını önlemektir. Özellikle veri tekrarının önlenmesi için geliştirilmiş olan ilişkisel veri tabanları, veri tabanları sistemlerinde bir çığır açmıştır. Günümüz dünyasında veri tabanı sistemlerini kullanmayan kurum ya da kuruluş yok denecek kadar azdır [10].

İstatistik: İstatistik ve veri madenciliği birbiriyle ilişkili olan iki disiplindir. Veri madenciliği daha sonra ortaya çıkan bir disiplin olduğu için veri madenciliği istatistikin bir alt dalı olarak da görülmektedir. Veri madenciliğini istatistikten ayıran en belirgin özellik, VM'nin veri tabanı sistemleri, makine öğrenimi ve bunun gibi birçok disiplinle ilişkili olmasıdır [35].

Makine Öğrenimi: Bilgisayarların olayları öğrenmesini sağlayan bir teknolojidir. Genel olarak örnek bir veri kümesi kullanılarak olayların giriş ve çıkışları arasında bağlantı kurulması sağlanır. Öğrenilen bilgilere benzeyen olaylar karşısında kararlar verilip ve yorumlar yapılarak problemler çözülür, böylelikle bilgisayar daha önceden bilmediği bir olay hakkında çözüm önerebilmektedir.

Veri Görselleştirme: Görsel görüntüler kullanıcılara veri tabanında tutulan verilerin karakteristikleri için genel bir bakış açısı kazandırabilir. Veri görselleştirme metodu, eldeki veriye ve kullanılan modele göre değişebilmektedir [26].

Diğer Disiplinler: VM yukarıda sayılan disiplinler haricinde veri ambarları, yapay sinir ağları ve genetik algoritmalar ile de ilişki içerisindedir.

2.4. Veri Madenciliği Uygulama Alanları

Veri madenciliği birçok disiplini kapsayan bir yaklaşımdır. Aslında yeni bir disiplin olmasına rağmen ilişkide olduğu alanlar çok fazla olduğu için uygulama alanı oldukça geniştir. Günümüzde işletmelerin tamamına yakını sahip oldukları müşterilerin hareketlerini kaydetmek ve bu müşterilerin geleceğe yönelik davranışlarının ne olacağı konusunda bilgi sahibi olmak isterler. VM işletmelerin bu

isteklerine cevap verebilen bir tekniktir. Bu yüzden veri madenciliğinin uygulama alanı oldukça geniştir. VM pazarlama, perakendecilik, bankacılık ve finans, ulaşım, tıp ve daha birçok farklı alanda kullanılmaktadır. Aşağıda veri madenciliğinin yaygın olarak kullanıldığı alanlar ele alınmıştır [36].

2.4.1. Pazarlama sektörü

Günümüzde serbest rekabet ortamında satış ve pazarlama faaliyetleri açısından çok önemli olan zaman ve piyasa verilerinin karar destek amaçlı bilgi haline dönüştürülmesi doğru ve zamanında kararlar alma, bu kararların geciktirilmeden hayata geçirilmesi ve işletmelerin varlıklarını devam ettirebilmeleri açısından çok önemlidir. Bu amaçla veri madenciliğinin yöntemlerinden biri olan karar ağaçları sayesinde böyle bir yapıya ulaşmak mümkündür. Uygulamada karar ağaçları özellikle karar destek sistemlerinde yaygın olarak kullanılmaktadır [26].

2.4.2. Perakende sektörü

Perakende sektöründe veri madenciliği, aşağıdaki uygulamalar için kullanılmaktadır:

Sepet Analizi: Müşterilerin alış veriş yaparken hangi ürünleri birlikte satın aldığını analiz eder ve elde edilecek olan bilgi mağaza raf düzenlemeleri ve promosyon stratejilerinin belirlenmesinde kullanılır [30].

Satış Tahminleri: Perakendeciler satış tahminlerini stok kontrolünde kullanırlar. Örneğin bir müşteri bugün alışveriş yaparsa, bir sonraki alışverişini ne zaman yapacağı sorusuna cevap aranır [36].

2.4.3. Hilekârlıkların tespiti ve yönetimi

Şirketler için en önemli hususlardan biri olan sahtekârlığa yönelik eğilimlerin izlenmesi VM teknikleriyle mümkündür.

Tespit edilmek istenen hilekârlıklar şu alanlarda görülebilir [37]:

- Kredi kartı dolandırıcılığı
- Sigorta dolandırıcılığı
- Kara para aklama
- Bilgisayar ve bilgisayar ağlarına izinsiz girme
- İletişim sektöründeki dolandırıcılıklar
- Abonelik dolandırıcılığı

2.4.4. Bankacılık ve finans sektörü

VM finans sektöründe işletme riskinin azaltılması, risk derecelendirme tahmini, maliyetlerin düşürülmesi, doğru ve etkin kredi kararı verebilme, müşteri ve çalışan memnuniyetinin artırılması gibi konularda kullanılmaktadır [38].

Chye ve Gerry tarafından bankacılık alanında "Churn Analizi" adında bir uygulama gerçekleştirmişlerdir. Bu uygulamanın temel amacı bir bankanın bugünkü müşterilerinin hesap hareketleri göz önünde bulundurularak gelecek altı ayda rakip firmalara geçip geçmeyeceğinin belirlenmesidir. Bu analiz dışında VM, bireysel kredi başvurularında yaş, cinsiyet, meslek, gelir ve gider kriterleri baz alınarak kişinin borcunu ödeyip ödeyemeyeceğinin tespitinin yapılmasını sağlayabilmektedir [39].

2.4.5. Sağlık

Sağlık alanında da veri madenciliği yaygın olarak kullanılmaktadır [40]. Yapılan araştırmalar sonucunda ana kütledeki anormal durumların ortaya çıkarılmasıyla, "koleranın tedavisi", "şizofreni ve kişinin doğduğu ay arasındaki bağ", "yüksek dozaj uyuşturucu alıp ölenler ve içinde bulunulan ay" arasındaki bağıntıların veri madenciliği ile tespit edildiği görülmüştür [41]. Örneğin Breault, Fos ve Goodall, Amerika'da yapmış oldukları bir araştırmada, diyabet hastalığının oluşmasına sebep

olan faktörleri veri madenciliğinin sınıflandırma tekniklerinden biri olan karar ağaçları ile belirlemişlerdir [42].

2.4.6. Telekomünikasyon

Telekomünikasyon sektörünün en önemli sorunu müşteri kaybıdır. Kuruluşlar hangi müşterilerinin rakip firmalara geçeceğini tespit edebilirlerse, bu müşterileri elde tutma amaçlı stratejiler geliştirebilir, düşük maliyetli ve etkili kampanyalar düzenleyerek bu sorunu çözme yoluna gidebilirler.

Telekomünikasyon sektöründe bulunan firmalar müşterilere ait ayrıntılı arama kayıtlarına sahiptirler. Bu firmalar bu verileri kullanarak, benzer özellik gösteren müşterileri bölümlendirip(müşteri segmantasyonu), fiyatlandırma ve promosyon stratejileri geliştirebilirler. Bu stratejiler hem kendi firmasıyla çalışan müşterilerin elde tutulması hem de rakip firmalardaki müşterilerin de kendi firmasıyla çalışmasını sağlayabilmektedir [36].

2.5. Veri Madenciliği Algoritmaları

Veri madenciliği algoritmaları genel olarak iki ana gruba ayrılır [43]:

Doğrulamaya Dayalı Algoritmalar: Bu tip algoritmalarda kullanıcı tarafından ispatlanmak istenen bir hipotez ortaya sürülür ve VM algoritmalarıyla bu hipotez ispatlanmaya çalışılır. Örnek olarak hipotez testi verilebilir.

Keşfe Dayalı Algoritmalar: Bu tip algoritmalarda doğrulamaya dayalı algoritmaların tersine, ortada ispatlanması istenen hipotezler yoktur. Bu algoritmalar otomatik keşfe dayanmaktadır. Bunlara örnek olarak istisnai durumların keşfi, karar ağacı, kümeleme gibi algoritmalar verilebilir.

2.5.1. Hipotez testi

Hipotez testi algoritmaları doğrulamaya dayalı algoritmalardır. Doğrulanacak hipotez örnek veri tabanı üzerindeki verilerle test edilir. Test işlemi bir uzman tarafından aşağıdaki ihtiyaçlardan dolayı yapılır [11]:

- Bir kural ortaya çıkarılmak istendiğinde,
- Ortaya çıkarılmış bir kuralın budanması veya genişletilmesinde.

2.5.2. Sınıflama algoritması

Kayıtların belli bir sınıfa alınması onların bazı ortak özelliklere sahip olmasını gerektirmektedir. Ortak özelliklere sahip olan kayıtların hangi özellikleriyle bu sınıfa girdiğini belirleyen algoritmaya sınıflama algoritması denir.

Sınıflama algoritmasının kullanım alanları sigorta risk analizi, banka kredi kartı sınıflaması, sahtecilik tespiti gibi alanlar örnek olarak gösterilebilir.

2.5.3. Kümeleme algoritması

Kümeleme algoritmasındaki amaç verilerin alt kümelere ayrılmasını sağlamaktır [44]. Sınıflama algoritmasında olduğu gibi ortak özelliklere sahip olan veriler bir kümeye dâhil edilir. Alt kümelere ayrılması kaydın hangi alt kümeye girdiği kümeleme algoritması aracılığıyla bulunur.

Kümeleme algoritması daha çok astronomi, nüfus bilimi, bankacılık uygulamaları, vb. uygulamalarda kullanılır [11].

2.5.4. Eşleştirme algoritması

Eşleştirme algoritması sınıflama algoritmasına benzemektedir [45]. Sınıflama algoritmaları ile eşleştirme algoritmaları arasındaki en önemli fark, eşleştirme

algoritmasında sınıflandırmada olduğu gibi bir sınıfa dâhil edilmesi amaçlanmamaktadır. Eşleştirmedeki amaç örnek veri kümesindeki nesnelerin nitelikleri arasındaki ilişkilerin belirlenmesidir. Böylelikle, nitelikler arasındaki bütün kombinasyonlar ortaya çıkarılarak bütün niteliklerin farklı kombinasyonlarındaki farklı değerleri denenerek örüntüler keşfedilmeye çalışılır [46].

3. SALDIRI TESPİT SİSTEMLERİ VE ÇÖZÜMLERİ

Sistemlerin güvenliğiyle ilgili tehditlerin sayısının ve türlerinin artmasıyla birlikte bilgi güvenliği teknolojilerinde de hızlı bir gelişim yaşanmaktadır. Bilgisayarların güvenliğini sağlamak, yetkisi olmayan kişilerin sistemlere girip bilgileri ele geçirmelerini veya değiştirmelerini engellemek için bu alanda ilk olarak kimlik doğrulama ve erişim kontrolü gibi güvenlik mekanizmaları geliştirilmiştir. Bu mekanizmalar güvenliğin birinci aşamasıdır. İnternetin yaygınlaşmasıyla birlikte bilgisayar sistemlerine yönelik yapılan saldırı tiplerinin çeşitliliğinde ve sayısında ciddi artışlar olmuştur. Tehditlerdeki bu artış karşısında bu saldırıları engellemek için yukarıdaki mekanizmalar dışında yeni mekanizmaların varlığına ihtiyaç duyulmuştur. Güvenlik mekanizmalarının ikinci aşaması olarak da ifade edilen bu mekanizmalar güvenlik duvarları(firewall), güvenlik tarayıcıları(vulnerability scanner) ve STS'lerden oluşmaktadır. Bu güvenlik mekanizmalarının tamamı bir bütündür. Bunlardan hiçbiri tek başına bir sistemin güvenliği için yeterli değildir. Güvenli bir sistem için bu yapıların birbirlerini destekleyecek şekilde beraber kullanılması gereklidir [47].

Sistemlerin güvenliğini sağlamak, sistemlere yapılan saldırıları engellemek için yapılan bu çalışmalar ile kurumsal güvenlik düzeyini artırmak amaçlanmaktadır. Bu çalışmalar üç temel kategoride toplanabilir. Bu alanlar [1];

- Güvenlik politikaları ve işlemler
- Teknoloji
- Eğitim ve bilgilendirme

Kurumsal güvenliğin artırılması için belirtilen alanların hepsinde çalışmaların yapılması zorunludur; alanların birinde eksik bırakılacak çalışmalar diğer alanlarda yapılan çalışmaların etkisini olumsuz bir şekilde etkileyecektir.

Yukarıda bahsedilen politikaların sağlanması kadar, ihlallerin tespiti de önem arz etmektedir. İhlallerin tespiti için monitorizasyon işlevini yerine getiren teknolojiler kullanılmaktadır. Saldırı tespitini sistemleri yetkisiz kullanma ya da yetkisi dahilinde bulunmayan işlemleri yapma girişiminde bulunan kişileri ya da programları tespit etme çalışması şeklinde ifade edebiliriz [48].

Bilgi güvenliği sağlanırken aslında önemli olan konulardan biri de var olan sistemin sürekliliğinin sağlanmasıdır. Bunu da sağlamak için yapılan saldırılara karşı alınan önlemlerin güncel olması gerekmektedir. Bu güncelliğin sağlanması ancak değişen saldırı tiplerinin ve yöntemlerinin bilinmesiyle sağlanabilmektedir. Ancak saldırı tipinin ve yönteminin bilinmesi için önce "saldırı"nın ne anlama geldiği bilinmesi gerekmektedir. Bu nedenle saldırı tespiti konusunda yapılan çalışmalarda saldırının birçok tanımı yapılmıştır [6]. Bu tanımlardan birini de Anderson şu şekilde yapmıştır. Bir saldırı, izin almadan bilgiye ulaşım, değiştirme, sistemi kullanılmaz ve güvenilmez hale getirmektir [49].

Anderson'un 1980'de saldırı için yapmış olduğu tanım hala geçerliliğini koruyan bir tanımdır. Günümüzdeki anlamıyla saldırıyı tanımlayacak olursak bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini tehlikeye atabilecek girişimlerin kümesidir denilebilir [50].

Saldırı tespiti ise bir bilgisayar sisteminde veya ağda meydana gelen olayları izleyerek, bilginin mahremiyetini, bütünlüğünü ve erişilebilirliğini bozmak için yapılan hareketlerin analiz edilerek bunların saldırı olduğunun tespit edilerek bu hareketlerin analiz edilmesi işlemidir. [51].

Adından da anlaşıldığı gibi saldırı tespiti işini yapmak için geliştirilen sistemlere "saldırı tespit sistemleri" denir. Saldırı tespit sistemlerinin literatürdeki bazı tanımları aşağıda sunulmuştur:

- Bilgisayar sistemlerine yapılan atakları ve kötüye kullanımları belirlemek için tasarlanmış sistemlerdir [52].
- Tercihen gerçek zamanlı olarak, bilgisayar sistemlerinin yetkisiz ve kötüye kullanımı ve suiistimalini tespit etmek için kullanılırlar [48].
- Saldırıyı durdurma girişiminde bulunmayan ve olası güvenlik ihlali durumlarında, sistem güvenlik çalışanlarına uyarı mesajı(alarm) veren sistemlerdir [53].
- Bilgisayar sistemlerinin kaynaklarına veya verilerine yetkisiz erişimleri belirler [54].
- Bilgisayar güvenliği alanındaki "hırsız alarm"lardır [55].
- Bilgisayar veya ağ sistemine yapılan yetkisiz erişimleri tespit etmek için kullanılan yazılım araçlarıdır. [56].

Saldırı tespit sistemlerinin kullanımı üç temel başlık altında özetlenebilecek faydalar sağlamaktadır [1]:

- Erken tespit: Saldırı tespit sistemlerinin en önemli faydalarından biri olan erken tespit yapabilmesidir. STS'ler başlayan bir ihlali sistem yöneticilerinden çok daha önce tespit edebilir ve olayla ilişkili sistem sorumlusunu SMS, e-posta ya da çağrı cihazı gibi farklı şekillerde anında uyararak ihlalin etkisi ciddi bir seviyeye ulaşmadan minimize edilebilir.
- Detaylı bilgi toplanması: STS'ler sayesinde sistem yöneticileri sürmekte olan ya da geçmişte gerçekleşmiş saldırı örüntüleri hakkında detaylı bilgi edinebilir. Bu bilgiler kullanılarak sistemin açıklıkları tespit edilip sistem tekrar konfigüre edilerek daha güvenli bir hale getirilebilir.
- Toplanan bilgilerin kanıt niteliği: Sistem tarafından toplanan bilgiler, hukuki yollara başvurulduğunda kanıt niteliğinde olabileceği gibi, başka kurumlardan kaynaklanan bir ihlalde ilgili kurumun yetkilileriyle bu bilgiler kullanılarak temasa geçilerek görüşmeler yapılabilir.

3.1. STS'lerin Tarihçesi

1970'lerden itibaren güvenlikle ilgili çalışmalar yapılmasına rağmen, saldırı tespit kavramı ilk olarak Anderson'un "Bilgisayar Güvenliği Tehdit Gözetleme ve İzleme (Computer Security Threat Monitoring and Survaillance)" makalesi ile 1980'de ortaya çıkmıştır [57].

STS'ler başlarda sadece basit bilgisayar sistemleri için düşünülmüştür. İkinci nesil STS'lerde ise veri madenciliğiyle saldırı tespitinin de özü olan "denetleme izi(audit trail)" kavramı günümüz STS'lerin vazgeçilmezidir. Böylelikle veri tabanı mantığının bilgi güvenliği alanındaki önemi ortaya çıkmıştır. Bunu izleyen çalışmalarda, denetleme izinin el ile değil de otomatik olarak elde edilmesine yönelik çalışmalar yapılmıştır [51]. 1985'ten beri bu yöndeki çalışmalar üzerinde özenle durulmuş ve statiksel yaklaşımlar temel alınarak saldırı tespit modelleri geliştirilmiştir. Bu çalışmalardan biri olan IDES(intrusion detection expertsystem), Denning tarafından geliştirilmeye başlanan ve özellikle 1988 – 1992 yılları arasında yapılan çalışmaların birçoğunu da üzerinde barındıran bir sistem olduğu için günümüz STS'lerinin gelişmesinde önemli bir rolü vardır [49]. Daha sonra adı NIDES(next-generation intrusion detection systems) olarak değiştirilen bu sistem, saldırı senaryolarından kural kümeleri çıkarılarak tasarlanmış kural tabanlı bir STS'dir. Denning çalışmasında 3 farklı istatistiksel model tanımlamıştır. Bu modeller;

- Kullanıcının belirli aralıklarla bir işlemi tekrar etmesine izin veren ve eşik değerine göre anormallik tespit eden model
- İstatistiksel momentlerin bilindiği varsayılarak tespit edilen sapmalar ile anormallik tespit eden model
- Anormalliklerin tek olaya değil birden çok olaya bağlı olduğu Markov modeli

olarak verilmektedir [52].

Günümüze kadar STS'lerin geliştirilmesinde statiksel yöntemler, kural tabanlı(rule based), eşik değeri belirleme(threshold value), durum geçiş diyagramları(state transition diagrams), yapay sinir ağları(artificial neural networks), veri madenciliği(data mining), yapay bağışıklık sistemi(artificial immune system), bulanık mantık(fuzzy logic) gibi farklı birçok yaklaşım uygulanmıştır [58,59].

STS'lerin geliştirilmesi kadar geliştirilen STS yazılımlarının test edilmesi de STS'nin başarımı açısından büyük önem teşkil etmektedir. STS yazılımlarının test edilmesinde "saldırı veri tabanları" örnekleri kullanılmaktadır [60-62]. STS'lerin temeli olan bu veri kümelerini oluşturmak oldukça zahmetli ve zaman alan bir iştir. Bu zahmetin dışında, oluşturulan verilerin gerçek verilerle uygunluğu da önemlidir. Günümüzdeki birçok uygulamada hala MIT (Massachusetts Institute of Technology)'nin Lincoln Laboratuarlarında 1998 yılında oluşturulan DARPA (Defence Advanced Research Projects Agency) saldırı tespit değerlendirme veri kümesi kullanılmaktadır [60]. DARPA her ne kadar 1998 yılında oluşturulsa da saldırı tespit değerlendirme veri kümesi olarak en kapsamlı olanıdır. Bu veri kümesinin amacı, saldırı tespit sistemlerinin başarımının ölçümüne katkıda bulunmak ve bu ölçümün belli bir standart dahilinde yapılmasını sağlamaktır. Bu yönde yapılan çalışmalar sürerken KDD'99(Knowledge Discovery and Data Mining) veri kümesi de saldırı tespit veri kümesi olarak DARPA verilerinin ön işlemden geçirilmesiyle elde edilmiştir [61]. KDD'99'un kullanımı DARPA verilerine göre daha kolay olmasına rağmen her iki veri tabanı da 2000 yılından sonra ortaya çıkan farklı saldırı türlerini içermemektedirler [6].

3.2. Saldırı Tipleri

Bilgisayar sistemlerine yapılan saldırılar, birçok araştırmacı tarafından farklı gruplandırmalar yapılmıştır [63,64]. Saldırganların sürekli kendilerini yenilemeleri ve bilgisayar sistemlerindeki açıkları tespit etmeleri nedeniyle saldırı tiplerindeki çeşitlilik çok fazla artmıştır. MIT Lincoln Laboratuarlarında yapılan bu çalışmalar sonucunda bilgisayar sistemlerine yapılan saldırılar kullandıkları yöntemlere göre

dört ana gruba ayrılmış ve DoS, U2R, R2L ve Probing olarak adlandırılmıştır [60]. Bu saldırıların detaylı tanımları aşağıdaki başlıklarda sunulmuştur.

Çizelge 3.1. STS'lerde kullanılan tekniklerin karşılaştırılması [65]

Teknik	**Tespit Yaklaşımı**	**Kullanılan Bilgi Kaynakları**	**Bilinen Ataklar**	**Bilinmeyen Ataklar**	**Performans**
İstatistiksel	Anormallik	Denetim verisi, kullanıcı profili	Evet	Evet	Orta
Veri Madenciliği	Anormallik	Denetim verisi, bilgi tabanı	Evet	Evet	Orta
Durum Geçiş Analizi	Kötüye Kullanım	Denetim kayıtları, bilinen atak örneklerinin	Evet	Hayır	Yüksek
Dosya Kontrol	Anormallik	Sistem dosyaları	Evet	Evet	Yüksek
Örüntü Eşleme	Kötüye Kullanım	Denetim kayıtları, saldırı imzaları	Evet	Hayır	Yüksek
Protokol Analizi	Anormallik	Denetim kayıtları, bir protokolün normal kullanım	Evet	Evet	Düşük
Tuş Vuruşu İzleme	Kötüye Kullanım	Bilinen saldırıların bilgi tabanı, Tuş vuruşları	Evet	Hayır	Yüksek

3.2.1. Hizmet aksattırma saldırıları

Hizmet aksattırma saldırıları olarak da bilinen DoS saldırıları sistemin hizmetlerini engellemek amacıyla yapılırlar. Bunu yapmak için sisteme cevap verebileceğinden çok daha fazla istek gönderilerek verilen hizmet aksattırılır. DoS saldırılarına örnek olarak SYN flood, Smurf, UDPstorm, Pingflood, Neptune, Mailbomb saldırıları verilebilir [66].

3.2.2. U2R saldırıları

U2R saldırıları sayesinde kullanıcılar normal yetkilere sahip olan kendi hesaplarından oturum açtıktan sonra yönetici yetkisine ulaşarak sistem üzerinde

istedikleri bilgilere erişebilirler. En çok bilinen U2R saldırı tipleri Eject, Ffbconfig, Fdformat, Loadmodule, Perl gibi saldırılardır [66].

3.2.3. R2L saldırıları

Bu saldırı tipinde saldırgan saldırdığı makineye ağ üzerinden paketler yollayarak açıkları tespit etmeye çalışır. Bu konuda birçok araç olması ve bu araçlara kolay erişilebilir olması nedeniyle, sistemde var olan açıklar kapatılmamışsa etkili ve kolay bir saldırı yöntemidir. En çok bilinen R2L saldırı tiplerine Dictionary, Guest, Imap, Named, Sendmail gibi saldırılar örnek olarak verilebilir [66].

3.2.4. Probing saldırıları

Probing saldırıları, ağı veya bilgisayarı tarayarak zayıflıkları tespit etmek ve sistem yapısıyla ilgili genel bir bilgiye ulaşmak için yapılmaktadır. Bu tip saldırılarda önce sistem hakkında detaylı bilgi edinildikten sonra saldırının nasıl olacağı belirlenir. Bu saldırılar için kullanılan araçlar aynı zamanda güvenlik uzmanları tarafından sistemin güvenliğinin test edilmesi için de kullanılır. En çok bilinen Probe saldırı tipleri, Ipsweep, Mscan, Nmap, Saint, Satan gibi saldırılar örnek olarak verilebilir [66].

3.3. Saldırı Tespit Sistemleri Türleri

Bir kaynağın bütünlüğünü, gizliliğini, güvenilirliğini veya erişilebilirliğini engelleme amaçlı yapılan tüm davranışlar saldırı niteliğindedir. STS'ler, bilgisayar sistemlerine karşı yapılan bu saldırıları saldırı gerçekleştiği anda ya da gerçekleştikten sonra tespit etmek, internetten veya intranetten gelebilecek, ağdaki sistemlere zarar verebilecek çeşitli paket ve verilerden oluşan bu tip saldırıları fark etmek ve bu saldırılara yanıt vermek üzere tasarlanmış sistemlerdir. Aslında saldırı tespit sistemlerini alarm sistemi olarak düşünebiliriz. STS'leri farklı kategorilere ayırmak mümkündür. Örnek olarak ISS(Internet Security Systems)'nin ortaya attığı modele göre STS'leri aktif ve pasif olarak gruplandırmıştır. İkinci bir sınıflandırmaya göre de anasistem-tabanlı(host-based) ya da ağ-tabanlı(network-based) şeklinde bir sınıflandırma

yapılmıştır. Üçüncü sınıflandırma ise bu iki sınıflandırmanın birleştirilmesiyle oluşmuştur. Buna göre aktif/anasistem-tabanlı, aktif/ağ-tabanlı, pasif/anasistem-tabanlı, pasif/ağ-tabanlı olarak da gruplandırılabilir. Burada bir sistemin aktif olması tespit edilen bir saldırıya gerçek zamanlı veya buna yakın cevap vermesi anlamındadır; pasif sistemlerde genelde saldırı kaydedilir ve daha sonra tekrar aynı örüntüye sahip bir saldırının yapılması durumunda saldırının olduğu tespit edilir ve bu sistem yöneticisine haber verilir. Saldırı tespitinin yapılmasında kullanılan analiz metotları iki tanedir. Bunlar kötüye kullanım saldırı tespiti ile anormallik saldırı tespitidir [47].

3.3.1. Kötüye kullanım saldırı tespiti

Bu yöntemde STS edindiği bilgileri kaydeder ve büyük boyutlu imza(signature) veri tabanlarıyla karşılaştırarak daha önceden belirlenmiş saldırılardan yola çıkarak güvenliği sağlamayı amaçlamaktadır.

Bu tip saldırı tespitinde başarım oranının yüksek olması için imza veri tabanlarının iyi bir şekilde tanımlanabilmesi ve veri tabanının büyüklüğünün iyi ayarlanması gerekmektedir. Veri tabanı gereğinden fazla büyük olursa yanlış alarm verme olasılığı büyürken, veri tabanındaki imzalar yetersiz kalırsa güvenlik sorunlarının ortaya çıkmasına sebep olabilir.

3.3.2. Anormallik saldırı tespiti

Anormallik saldırı tespiti(anomaly detection) yaklaşımının temeli, sistemde meydana gelen anormal olayları normal olaylardan ayırt etmeyle ilgilidir. STS için anormalliğin anlamı normal davranıştan sapma olarak nitelendirilmektedir. Normal davranışın elde edilmesi için sistemin uzun süre analiz edilmesi, incelenmesi gerekmektedir. Sistemdeki kullanıcı veya kullanıcı gruplarının davranışlarının belirlenmesi anormallik saldırı tespiti için en önemli iştir. Çünkü normal davranış profili belirlendikten sonra farklılık gösteren davranışlar saldırı olarak

nitelendirilecektir. Bu anlamda normal davranış profili ne kadar doğru belirlenirse saldırı tespitindeki başarım oranı o kadar artacaktır [6].

Bu yöntemde her kullanıcı ve kullanıcı grupları için ayrı ayrı profiller belirlenir. Bu yöntemin bir diğer adı profil tabanlı saldırı tespitidir. Bu profillerin oluşturulması dinamik bir şekilde veya elle yapılabilir. Normal kullanıcı için taban çizgisi(baseline) adı verilen bir kıstas belirlenir. Eğer ağda yapılan bir işlem taban çizgisinden çok fazla sapma göstermesi durumunda alarm tetiklenir [47].

3.4. STS'nin Önemi

Günümüzde her gün binlerce sistem saldırıya uğramaktadır. Bu saldırıların çeşitliliği çok fazla olduğu için bu saldırıların karakteristikleri hakkında herhangi bir bilgi elde edilememektedir. Bu durum saldırıların engellenmesi olasılığını da çok güçleştirmektedir. Saldırganların bir sisteme saldırırken kullandıkları iki yol vardır. Bunlardan birincisi, bilgisi fazla olmayan saldırganların tercih ettiği bazı otomatize edilmiş araçlarla yapılan saldırılardır. İkinci yolu tercih edenler ise bilgisi uzman seviyesinde olan saldırganlar tarafından saldırılan hedefe göre değişik yöntemler kullanılarak yapılan saldırılardır. İlk tip saldırıları önlemek ikinci tip saldırılara göre daha kolaydır. Çünkü otomatize edilerek bu işleri yapan araçların çalışma mantaliteleri bellidir. Bunları önlemek için bu çalışma mantıkları incelenir ve buna yönelik önlemler alınır; ancak ikinci tip saldırılarda daha savunmasız kalınır. Bunların belirgin bir hareket veya tarzı yoktur, su gibi bardağın şeklini alarak şekilden şekle girebildikleri için saldırganların engellenebilirliği çok düşük seviyededir. Bu tip saldırıların önlenmesi için geliştirilen yöntemler de çok çeşitlidir. Bu anlamda her ne kadar yerel ağ ne kadar karmaşık olursa veya saldırı önlemeye yönelik ne kadar bilinmedik yöntemler kullanılırsa saldırıların o kadar zor gerçekleşeceğine dair bir fikir oluşmuşsa da, bu fikir ikinci tip saldırı yöntemini kullanan saldırganlar için geçerli değildir [68].

Birinci tip saldırganları önlemeye yönelik olan firewall, anti virüs gibi sistemler aslında saldırganları birazcık da olsa yavaşlatmaktadırlar. Bu sistemler saldırganları yavaşlatma amaçlı kurulmuş olup düzenli olarak saldırı kayıtlarının incelenmesi yoluyla sistemin pasif olarak korunmasını sağlamaktadırlar. Bu durumda firewall ve anti virüs sistemlerine ek olarak aktif olarak da korunmayı sağlamak için STS'leri de kurmak gereklidir.

STS ile ağa yapılabilecek saldırıları belirleme ve gerektiği durumda engelleme imkanı sağlanmaktadır. Düzenli olarak tutulan saldırı kayıtları incelenerek saldırganlar ve saldırılar hakkında detaylı bilgiler elde edilebilir ve gerektiğinde kötü niyetli isteklerin ağa girmesine engel olunabilir [68].

4. VERİ MADENCİLİĞİ İLE BİLGİSAYAR AĞLARINDA YENİ BİR SALDIRI TESPİT ALGORİTMASI

Bu çalışmada Quinlan'ın ID3 ile karar ağacı oluşturma yaklaşımı göz önüne alınarak veri madenciliğinin sınıflandırma yöntemlerinden biri olan karar ağacı yöntemiyle yeni bir saldırı tespit algoritması geliştirilmiştir. Geliştirilen algoritma belli formatta toplanmış saldırı veri yığınından karar ağacı oluşturma ve bu karar ağacının test edilip saldırı tespit başarım oranının raporlanması gibi işlemleri yerine getirmektedir.

Uygulamada kullanılan saldırı verileri 42 nitelikten oluşmaktadır. Niteliklerin sonuncusu saldırı verisinin hangi tür saldırı tipinden olduğunu belirlemektedir. Bu çalışmada, standart ID3'ün ve önerilen yeni yöntemin karar ağacı oluşturma yaklaşımına göre oluşturulan karar ağaçlarının test verileriyle test edilip başarımları raporlanmaktadır. Ayrıca, oluşturulan ağacın açık kaynak bir araç olan Graphviz ile görselleştirilmesi sağlanmıştır. Örnek bir karar ağacı Ek-2'de sunulmuştur. Geliştirilen yazılım KDD Cup'99 veri kümeleri üzerinde denenmiş, bu deneyin sonuçları kaydedilmiş ve değerlendirilmiştir.

4.1. STS Veri Kümeleri

Bazı uygulama geliştiriciler, STS çalışmalarında eğitim ve test veri kümesi olarak kullanacakları veri kümelerini kendileri oluşturmuşlardır. Ancak, bu oldukça zahmetli ve zaman alan bir çalışmadır. STS geliştiricilerinin sınırlı olan zamanını veri kümesi oluşturmaya harcaması iş gücünü arttırmaktadır. Diğer yandan, güncel ve standart olmayan bu veri kümeleriyle geliştirilen sistemde başarı oranı yüksek çıksa da bu sonuçların gerçeklikten uzak olduğu söylenebilir. Bu anlamda, sistemler için standart bir eğitim ve test ortamının oluşturulması için güncel ve standart hale getirilmiş STS veri kümelerine ihtiyaç duyulmaktadır [6].

Literatürde, bu kapsamda 1998-2000 yılları arasında yoğun bir çalışma yapılmıştır [60,61,70]. Ancak, bu çalışmalar sonucunda oluşturulan veri kümeleri gerçeğe uygun

ve güncel olmadığı için daha yoğun bir şekilde çalışılarak STS'ler için standart ve güncelliğini koruyan kümelerinin oluşturulması gerekmektedir [6].

Bu konuda daha önce yapılmış ve günümüzde de kullanılmaya devam eden veri kümeleri arasında IDEVAL(Intrusion Detection Evaluation Dataset) ve KDD'99 hakkında detaylı açıklamalar 4.1.1 ve 4.1.2'de verilmiştir.

4.1.1. IDEVAL

IDEVAL adı verilen ilk standart veri kümesi, MIT Lincoln Laboratuarları, Bilgi Sistemleri Teknolojisi grubunun, DARPA ve AFRL(Hava Kuvvetleri Araştırma Laboratuarı - Air Force Research Projects Agency) desteğiyle yürüttüğü çalışmalar sonucunda, saldırı tespit sistemlerinin değerlendirilmesi ve güncel saldırılarla karşılaştırılarak başarımının raporlanması için oluşturulmuştur. Bu çalışma saldırı tespit sistemlerinin başarımının değerlendirilmesi kapsamında ilk standart veri kümesidir [6].

IDEVAL veri kümeleri

Lincoln laboratuarlarında saldırı tespit sistemlerinin başarımlarını ölçmek için 1998-2000 yıllarında arasında yoğun çalışmalar yapılmıştır. DARPA saldırı tespit değerlendirme çalışmaları kullanılarak 1998 ve 1999 DARPA IDEVAL veri kümeleri elde edilmiştir [61,71]. Bu veri kümeleri sayesinde, saldırı tespit sistemlerinin gerçek zamanlı olmayan(off-line) ve gerçek zamanlı olan iki farklı değerlendirmesi yapılabilmektedir [6].

1998 DARPA

1998 DARPA veri kümesi, 1998 IDEVAL veri kümesinin gerçek zamanlı olmayan kısmı için geliştirilmiştir [61].

1999 DARPA

DARPA 1999 veri kümesi, 1998'deki veri kümesi kullanılarak yeni ve farklı bazı saldırıların da veri kümesine dahil edilmesi yoluyla oluşturulmuştur [72]. DARPA veri kümelerinin esas amacı STS'lerin değerlendirilmesini sağlamaktır [6].

4.1.2. KDD'99

KDD'99 veri kümesi DARPA veri kümesinin bazı önişlemlerden geçirilmesiyle elde edilen 41 özelliğe sahip bir veri kümesidir. Bu veri kümesinin amacı diğer veri kümelerinde olduğu gibi farklı STS'lerinin eğitim ve test işlemlerini kolaylıkla gerçekleştirmek ve bu sistemlerin değerlendirilmesini sağlamaktır. DARPA ile her ne kadar STS'ler için veri kümesi probleminin çözülmesi adına önemli adımlar atılmışsa da KDD'99 ile eğitim ve test sonuçları çok daha hızlı alınabilmesi ve buna bağlı olarak STS'lerin değerlendirilmesinin daha kısa sürede yapılabilmesi adına bu alanda atılmış önemli bir adımdır [6].

KDD'99 veri kümesi, 9 temel ve 32 adet türetilmiş olmak üzere toplam 41 özellikten oluşmaktadır. Bu 41 özellik aşağıdaki gibi 3 ana gruba ayrılmıştır:

- İçerik özellikleri (content features)
- Sunucu tabanlı trafik özellikleri (host-based traffic features)
- Zamana bağlı trafik özellikleri (time-based traffic features)

Çizelge 4.1, Çizelge 4.2 ve Çizelge 4.3'de, sırasıyla bu 3 grup ve gruplara ait veri özellikleri gösterilmiştir.

Çizelge 4.1. İçerik özellikleri

Özellik adı	**Tanım**	**Tip**
Duration	Bağlantı uzunluğu	sürekli
protocol_type	Protokol tipi	ayrık
Service	Servis tipi	ayrık
src_bytes	Kaynaktan hedefe veri	sürekli
dst_bytes	Veri byte sayısı	sürekli
Flag	Bayrak	ayrık
Land	Kaynak ve hedef IP aynı ise 1 değilse 0	ayrık
Wrong_fragment	Yanlış parçalama	sürekli
Urgent	Acil paket sayısı	sürekli

İçerik özellikleri, sadece TCP bağlantılarıyla ilgili olan özelliklerdir. Diğer gruplarda olduğu gibi ağdaki veriler üzerinde herhangi bir önişlem yapılmasına gerek yoktur [6]. Sunucu tabanlı trafik özellikleri, etki alanı (domain) bilgisi ile ilgili içerik özellikleridir.

Çizelge 4.2. Sunucu tabanlı trafik özellikleri

Özellik adı	**Tanım**	**Tip**
Hot	“hot” göstergesi	sürekli
Num_failed_logins	Hatalı giriş sayısı	sürekli
Logged_in	Giriş başarılı ise 1 değilse 0	ayrık
Num_compromised	Gizliliğin ihlal edilme sayısı	sürekli
root_shell	“Root Shell” elde edildiyse 1 değilse 0	ayrık
su_attempted	“Su Root” komutu girildiyse 1 değilse 0	ayrık
Num_root	“Root” erişim sayısı	sürekli
Num_file_creations	Dosya oluşturma işlemleri sayısı	sürekli
Num_shells	Shell promptlarının sayısı	sürekli
Num_access_files	Kontrol dosyalarına erişim işlemleri sayısı	sürekli
Num_outbound_cmds	ftp oturumunda giden komut sayısı	sürekli
is_hot_login	Giriş “hot” listesindeyse 1 değilse 0	ayrık
is_guest_login	Giriş “guest” ise 1 değilse 0	ayrık

Zamana bağlı trafik özellikleri, "aynı sunucu" ve "aynı servis" özelliklerine göre kullanılan özelliklerdir. "Aynı sunucu" özellikleri, son iki saniye içerisinde aynı sunucuya yapılan bağlantılardır, "aynı servis" özellikleri ise son iki saniye içerisinde aynı servise yapılan bağlantılardır [6].

Çizelge 4.3. Zamana bağlı trafik özellikleri

Özellik adı	**Tanım**	**Tip**
Count	Aynı sunucuya önceki iki bağlantıyla aynı bağlantıların sayısı	sürekli
serror_rate	"SYN" hata bağlantılarının yüzdesi	sürekli
rerror_rate	"REJ" hata bağlantılarının yüzdesi	sürekli
same_srv_rate	Aynı servise bağlantıların yüzdesi	sürekli
Diff_srv_rate	Farklı servislere bağlantıların yüzdesi	sürekli
Srv_count	Aynı servise önceki iki bağlantıyla aynı bağlantıların sayısı	sürekli
Srv_serror_rate	"SYN" hata bağlantılarının yüzdesi	sürekli
Srv_rerror_rate	"REJ" hata bağlantılarının yüzdesi	sürekli
Srv_diff_host_rate	Farklı servislere bağlantıların yüzdesi	sürekli

KDD'99 veri kümesi 38 farklı atak içermektedir. KDD'99 eğitim veri kümesinde bulunan 24 atak ve bu atakların ait oldukları saldırı tipleri ve saldırıları veri kümesinde bulunduran örnek sayıları Çizelge 4.4'de verilmiştir.

Çizelge 4.4. KDD'99 veri kümesinin %10'luk kısmındaki saldırı örneklerinin sayıları

Atak	**Örnek sayısı**	**Kategori**
smurf.	280790	DoS
neptune.	107201	DoS
back.	2203	DoS
teardrop.	979	DoS
pod.	264	DoS
Land.	21	DoS
normal.	97277	normal
satan.	1589	probe
ipsweep.	1247	probe
portsweep.	1040	probe
nmap.	231	probe
warezclient.	1020	R2L
guess_passwd.	53	R2L
warezmaster.	20	R2L
İmap.	12	R2L
ftp_write.	8	R2L
multihop.	7	R2L
Phf.	4	R2L
Spy	2	R2L
buffer_overflow	30	U2R
rootkit.	10	U2R
loadmodule	9	U2R
perl	3	U2R

Sadece test veri kümesinde yer alan, eğitim kümesinde yer almayan ve etiketlenmiş KDD'99 dosyasından alınan 14 farklı atağa ait saldırı tipi ve örnek sayıları Çizelge 4.5'de sunulmuştur.

Çizelge 4.5. Eğitim kümesinde yer almayan test veri kümesinde bulunan ataklar

Atak	Örnek sayısı	Kategori
apache	794	DoS
mailbomb	5000	DoS
processtable	759	DoS
udpstorm	2	DoS
mscan	1053	probe
saint	736	probe
httptunnel.	138	R2L
named	17	R2L
sendmail	17	R2L
snmpgetattack	1040	R2L
xlock	9	R2L
xsnoop	4	R2L
ps	16	U2R
xterm	13	U2R

4.2. Standart ID3 Algoritması

Bu bölümde standart ID3 algoritmasının temeli olan karar ağaçları, karar ağaçlarıyla sınıflandırma ve ID3 algoritmasının çalışma prensipleri üzerinde durulmuştur.

4.2.1. Karar ağaçları

Karar ağaçları bilgi yığınının daha düzenli olarak ifade edilmesini sağlayan bir yapıdır. Günümüzde karar ağacı yapısı birçok problemin çözümünde kullanılmaktadır. Temel olarak karar ağacında üç birim vardır:

- Düğüm
- Dal
- Yaprak

Düğüm soruları, dal bu soruların cevaplarını, yaprak ise kararın verildiği sınıfı temsil etmektedir. Buna göre ağacın ilk düğümünden itibaren sorular sorulmaya başlanır ve

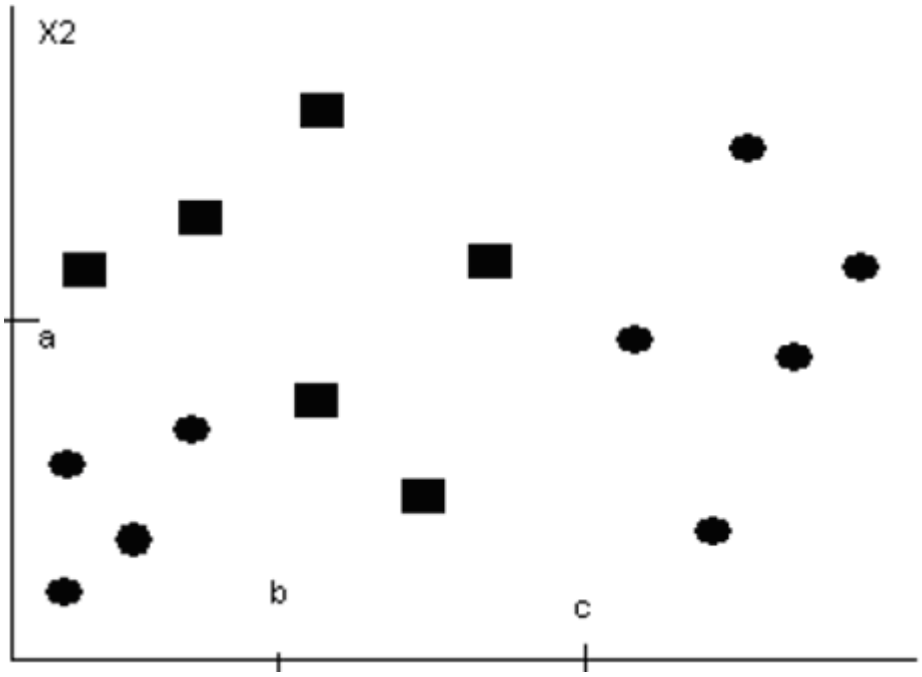
alınan cevaplara göre yapraklara gelene kadar sorular sorulmaya devam edilir. Aslında karar ağaçları if-then yapısıyla ifade edilen bilgi yığınının bir arada toplanmış halidir. Karar ağacının yapısı her ne kadar bu kadar basit bir şekilde ifade edilse de karar ağacının oluşturulması zor bir işlemdir. Şimdiye kadar karar ağacı oluşturmaya yönelik birçok algoritma geliştirilmiştir. Bu algoritmalara geçmeden önce karar ağacının türleri hakkında bilgi vermek yararlı olacaktır. Düğümlerin yapısı bakımından karar ağaçlarının üç çeşit tipi vardır:

- Tek değişkenli karar ağaçları
- Çok değişkenli karar ağaçları
- Melez karar ağaçları

Tek değişkenli karar ağaçlarında, düğümlerde sorulan sorular ilgili olayın bir tek değişkenine bakılır, çok değişkenli karar ağaçlarında, düğümlerdeki sorular birden fazla değişkenin değerine bakılırken, melez ağaçlarda ise bu iki test biçimi de kullanılır. Ancak bu yöntemlerden biri diğerinden daha iyidir diyemeyiz; her iki yöntemin de birbirine göre farklı avantajları ve dezavantajları bulunmaktadır [73].

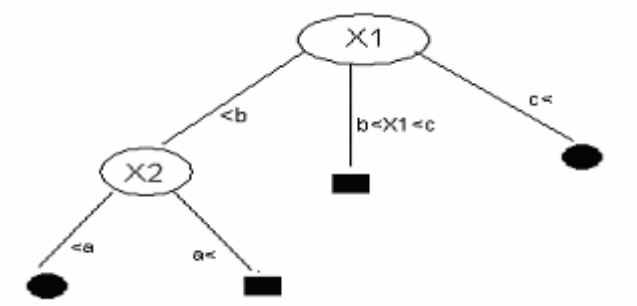
Karar ağacı oluşturma algoritmalarında olay kümesi belli özelliklere sahiptir. Bu özelliklerin aldığı değerler ya kategoriktir ya da sayısaldır (Örnek: hava(soğuk, ılık, sıcak): KATEGORİK ya da NOMİNAL, hava(63,9, 88,8): SAYISAL). Bu özelliklerin aldığı değerlere göre bunlara karşılık gelen sınıf oluşturulur [73].

Örnek olarak Şekil 4.1'de x_1 ve x_2 özelliklerine sahip bir olaylar kümesinin sonuçları vardır.



Şekil 4.1. Koordinat sistemleriyle ifade edilen olay kümeleri [73]

Bu sonuçlar kare ve daire olmak üzere iki sınıftan ibarettir. Bu şekle göre Şekil 4.2'deki ağaç oluşturulur.



Şekil 4.2. Şekil 4.1'deki veri kümesi ile oluşturulan ağaç [73]

Şekilden de görüldüğü gibi bu karar ağacı tek değişkenli bir karar ağacıdır. Burada olayların tek bir özelliği göz önüne alınmıştır.

Bu ağaçta aslında gizlenmiş dört adet farklı kural vardır. Bu kurallar aşağıdaki gibi yazılabilir:

X1 < b ve X2 < a => daire

X1 < b ve X2 > a => kare

b < X1 < c => kare

c < X_1 => daire

4.2.2. Sınıflandırma

Verinin içerdiği ortak özelliklere göre ayrıştırılması işlemi sınıflandırma olarak adlandırılır; veri madenciliğinde önemli bir konudur. Birçok sınıflandırma yöntemi vardır; karar ağaçları bunlardan birisidir. Karar ağaçları oluşturmak için temel olarak entropiye dayalı algoritmalar, sınıflandırma ve regresyon ağaçları, bellek tabanlı sınıflandırma modelleri biçiminde birçok yöntem geliştirilmiştir. ID3 ve C4.5 algoritmaları entropiye dayalı karar ağaçları oluşturma yöntemleridir [74].

Sınıflandırma bir öğrenme algoritmasına dayanır. Tüm veriler kullanılarak eğitme işlemi yapılmaz; veri topluluğuna ait bir örnek veri üzerinde gerçekleştirilir. Öğrenmenin amacı bir sınıflandırma modelinin yaratılmasıdır. Bir başka deyişle sınıflandırma, hangi sınıfa ait olduğu bilinmeyen bir kayıt için bir sınıf belirleme sürecidir. Örnek olarak, basit bir sınıflandırmayla müşteriler iki belirgin sınıfa ayrıştırılabilirler: "Ödemeleri üç gün içinde yapanlar" ve "ödemeleri üç günden sonra yapanlar" [74].

4.2.3. Sınıflandırma süreci

Sınıflandırılma süreci iki aşamadan oluşur.

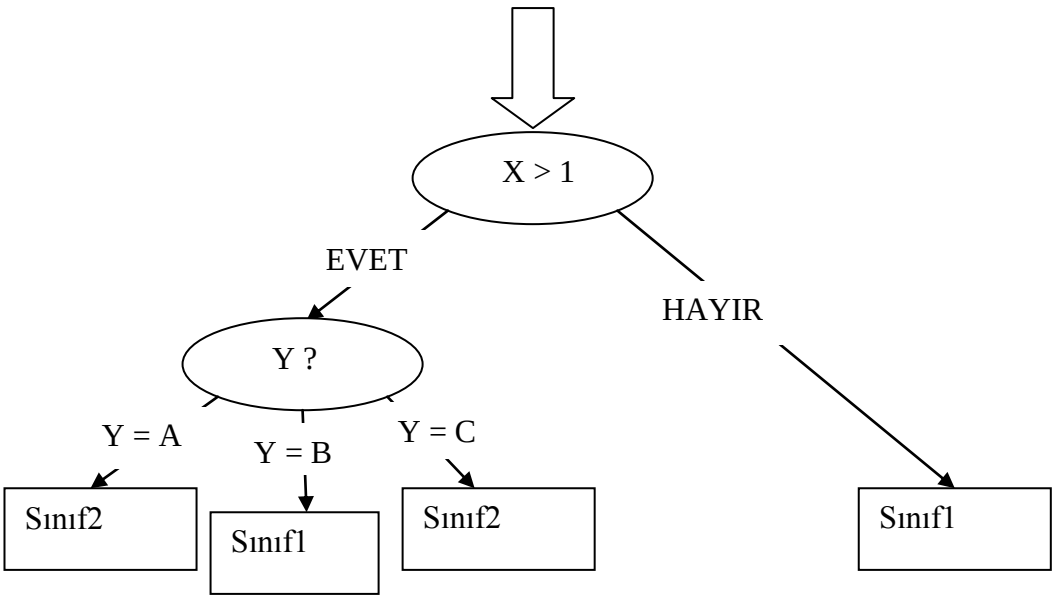
- İlk aşamada, eğitim veri kümesi kullanılarak uygun bir model ortaya konulur. Söz konusu model, veritabanındaki kayıtların nitelikleri(attribute) kullanılarak gerçekleştirilir ve bu modelin elde edilmesinde eğitim kümesindeki veriler kullanılır. Eğitim kümesi de veritabanındaki verilerden rasgele seçilir ve bu eğitim verileri üzerine karar ağacı oluşturma algoritmalarından biri uygulanır ve sınıflama modeli elde edilir.

- İkinci aşamada ise eğitim verileri kullanılarak ortaya konulan sınıf modeli test verileriyle test edilir. Karar ağacının amacı karar niteliğini belirlemektir. Test kısmında karar niteliği belli olan bir kayıtın sınıf modeline karar niteliği dışındaki bütün niteliklerin değerleri giriş olarak verilip karar niteliğinin değeri elde edilir ve bu değer test verisindeki karar niteliğinin değeriyle karşılaştırılır.

4.2.4. Karar ağaçlarıyla sınıflandırma

Karar ağaçlarında her bir nitelik bir düğümle ifade edilmektedir. Daha önceki bölümlerde de ifade edildiği gibi düğümler, dallar ve yapraklar ağaç yapısının elemanlarıdır. Şekil 4.3.'te örnek bir karar ağacı yer almaktadır.

X>1 ve Y=B değerlerini taşıyan örnekler sınıf1'de; Y=A ve Y=C koşullarına uygun olanlar sınıf2'de yer almaktadır. Ancak Y'nin değerini göz önüne almadan X≤1 koşuluna uygun örnekler sınıf1'de yer almaktadır [73].



Şekil 4.3. X ve Y nitelikleri üzerine uygulanan testleri içeren basit bir karar ağacı [73]

4.2.5. Karar ağaçlarında dallanma kriterleri

Karar ağaçlarında en önemli sorun bölümlemenin veya bir başka deyişle dallanmanın hangi kriter temel alınarak yapılacağıdır. Her farklı kriter için bir karar ağacı algoritması karşılık gelmektedir. Söz konusu algoritmaları aşağıdaki gibi gruplayabiliriz [74]:

- Entropiye dayalı algoritmalar
- Sınıflandırma ve regresyon ağaçları
- Bellek tabanlı sınıflandırma algoritmaları

Entropiye dayalı bölümlemeyi kullanan algoritmalara örnek olarak ID3 ve C4.5 algoritmaları, sınıflandırma ve regresyon ağaçları konusunda ise Twing ve Gini algoritmaları, bellek tabanlı sınıflandırma yöntemleri arasında k-en yakın komşu algoritması örnek olarak verilebilir [74]. Bu çalışmada entropiye dayalı algoritmalardan ID3 ve C4.5 algoritması üzerinde durulacaktır.

4.2.6. ID3 algoritması

Karar ağaçları yardımıyla sınıflandırma işlemlerini yerine getirmek üzere Quinlan tarafından birçok algoritma geliştirilmiştir. Bunlar arasında yer alan ID3 algoritması entropi tabanlı bir algoritmadır.

Entropi

Bir sistemdeki belirsizliğin ölçüsüne "entropi" denir. Örneğin, S bir kaynak olsun. Bu kaynağın $\{m_1, m_2, \ldots\ldots, m_n\}$ olmak üzere n mesaj ürettiğini varsayalım. Tüm mesajlar birbirinden bağımsızdır ve m_i mesajlarının üretilme olasılıkları p_i'dir.

$P = \{p_1, p_2, \ldots., p_n\}$ olasılık dağılımına sahip mesajları üreten S kaynağının entropisi H(S) şu şekilde hesaplanır [74]:

$$H(S) = - \sum_{i=1}^{n} p_i \log_2 p_i \qquad (4.1)$$

Örnek 1

Bir deneyin sonuçlarını içeren olasılık cetvelini göz önüne alacak olursak [74]:

Deney Sonuçları(S)	**a_1**	**a_2**	**a_3**
p_i	1/2	1/3	1/6

S = {a_1, a_2, a_3} deney kümesini ifade etsin. Söz konusu a_1, a_2 ve a_3 olaylarının belirsizliklerini sırasıyla hesaplayalım:

$-1/2\log_2 1/2$, $-1/3\log_2 1/3$, $-1/6\log_2 1/6$

Bu durumda toplam belirsizlik, yani entropi şu şekilde olacaktır:
$H(S) = -(1/2\log_2 1/2 + 1/3\log_2 1/3 + 1/6\log_2 1/6)$
$H(S) = 1.4591$

Örnek 2

Aşağıdaki sekiz elemanlı S kümesi için [74]:
S= {evet, evet, hayır; hayır; hayır; hayır; hayır; hayır}

Olasılıklar iki adet “evet” değeri için,
$p_1 = 2 / 8 = 0{,}25$,

diğer altı adet “hayır” değeri için,
$p_2 = 6 / 8 = 0{,}75$

değerleri hesaplanır. O halde $P = (p_1, p_2) = (0{,}25, 0{,}75)$ yazılır ve bu değerlere bağlı olarak S için toplam entropiyi aşağıdaki gibi elde edebiliriz.

$H(S) = -(p_1 \log_2 p_1 + p_2 \log_2 p_2)$

$H(S) = -(0{,}25 \log_2 0{,}25 + 0{,}75 \log_2 0{,}75)$

$H(S) = 0{,}81128$

Karar ağaçlarında entropi

Karar ağaçlarının oluşturulması esnasında dallanmaya veya bölümlemeye hangi nitelikten başlanılacağı çok önemlidir. Çünkü sınırlı sayıda kayıttan oluşan eğitim kümesi kullanılarak olası tüm ağaç yapılarını ortaya çıkarıp ve içlerinden en uygun olanını seçip ondan başlamak hiç de kolay bir şey değildir. Örneğin 5 nitelik ve 20 elemanlı bir eğitim kümesi kullanılarak çizilebilecek karar ağaçlarının sayısı 10^6'dan daha büyüktür. Bu yüzden karar ağacı algoritmalarının çoğu başlangıçta bazı önişlemler yaparak karar ağacı oluşturma yoluna giderler. Bunun için de ağacın hangi niteliğe göre dallanacağını belirlemek için entropi kavramı kullanılır. Daha önce de değinildiği gibi entropi bir veri kümesindeki belirsizliğin ölçüsüdür. Belirsizlik ne kadar fazla ise entropinin değeri de o kadar büyük olur. Örneğin, herkes Fenerbahçeli olsaydı, kime sorarsak soralım alacağımız yanıt hep aynı olacağı için entropi bu durumda sıfır olacaktır. Bütün olasılıkların eşit olması durumunda ise entropi değeri maksimum olacaktır [75].

Veritabanından eğitim için elde edilen kayıt kümesini göz önüne alalım. Eğitim kümesi sınıf niteliğinin alacağı değerlere göre $\{C_1, C_2, \ldots, C_k\}$ olmak üzere k sınıfa ayrıldığını düşünelim. Bu sınıflarla ilgili olarak ortalama bilgi miktarı gerekli olabilir. Burada T sınıf değerlerini içeren küme için P_T sınıfların olasılık dağılımını göstermektedir ve hesabı şu şekilde yapılır [74]:

$$P_T = \left(\left|\frac{C_1}{T}\right|, \left|\frac{C_2}{T}\right|, \left|\frac{C_3}{T}\right| .. \left|\frac{C_k}{T}\right| \right) \tag{4.2}$$

$|C_i|$ ifadesi C_i kümesindeki elemanların sayısını göstermektedir. Burada örneğin $p_1 = |C_1/T|$ olasılığının değerini ifade etmektedir. O halde T için ortalama bilgi miktarı veya bir başka deyişle entropi hesabı şu şekilde ifade edilir [74]:

$$H(T) = \sum_{i=1}^{n} p_i \log_2 p_i \qquad (4.3)$$

Örnek 1

Aşağıdaki on elemanlı RİSK kümesini göz önüne alalım [74].
RİSK = {var, var, var, yok, var, yok, yok, var, var, yok}
Bu küme için olasılıklar dağılımı yardımıyla entropiyi hesaplayalım. Burada C_1 sınıfı "var", C_2 sınıfı ise "yok" değerlerini içersin. Bu durumda,

$|C_1| = 6$
$|C2| = 2$

olduğuna göre, olasılıklar $p_1 = 6/10$, $p_2 = 4/10$ biçiminde hesaplanır. O halde olasılık dağılımı şu şekilde yazılabilir:
$P_{RİSK} = (6/10, 4/10)$
$H_{RİSK} = -(6/10\log_2 6/10 + 4/10\log_2 4/10)$
$H_{RİSK} = 0{,}97$

<u>Dallanma için niteliklerin seçilmesi için kazanç ölçütü</u>

T hedef niteliğini ifade etsin. Hedef niteliği olmayan yani sınıf niteliği olmayan bir X niteliğinin değerine bağlı olarak T kümesini T1, T2, …..,Tn alt kümelerine ayıracak olursak, T'nin bir elemanının hangi sınıfa ait olduğunu belirlemek için gerekli bilgi T_i'nin bir elemanının sınıfının belirlenmesi için gerekli olan bilginin ağırlıklı ortalamasıdır. O halde, buna göre T'nin bir elemanının sınıfını belirlemek için gerekli bilgiyi şu şekilde hesaplayabiliriz [74]:

$$H(X, T) = \sum_{i=1}^{n} \left| \frac{T_i}{T} \right| H(T_i) \quad (4.4)$$

Ağacın dallanması için niteliklerin seçilmesinde kullanılan "kazanç ölçütü" adı verilen kavram şu şekilde tanımlanır:

$$\text{Kazanç}(X, T) = H(T) - H(X, T) \quad (4.5)$$

Burada ayırma işlemi yapılırken Kazanç(X, T) değerinin maksimum olması amaçlanır. Böylelikle kazancı maksimize edebilecek X testi seçilmiş olur.

Örnek 1

Aşağıdaki eğitim verilerini göz önüne alalım. Bu verilerden yararlanarak BORÇ niteliği için kazanç ölçütünü hesaplayacağız [74].

Çizelge 4.6. Eğitim Kümesi [74]

Müşteri	**Müşteri**	**Borç**	**Gelir**	**Risk**
1	YÜKSEK	YÜKSEK	İŞVEREN	KÖTÜ
2	YÜKSEK	YÜKSEK	ÜCRETLİ	KÖTÜ
3	YÜKSEK	DÜŞÜK	ÜCRETLİ	KÖTÜ
4	DÜŞÜK	DÜŞÜK	ÜCRETLİ	İYİ
5	DÜŞÜK	DÜŞÜK	İŞVEREN	KÖTÜ
6	DÜŞÜK	YÜKSEK	İŞVEREN	İYİ
7	DÜŞÜK	YÜKSEK	ÜCRETLİ	İYİ
8	DÜŞÜK	DÜŞÜK	ÜCRETLİ	İYİ
9	DÜŞÜK	DÜŞÜK	İŞVEREN	KÖTÜ
10	DÜŞÜK	YÜKSEK	İŞVEREN	İYİ

Hedef nitelik olan RİSK niteliğinin sınıf değerlerini şu şekilde gösterelim.

RİSK = {kötü, kötü, kötü, iyi, kötü, iyi, iyi, iyi, kötü, iyi}

RİSK kümesinde "kötü" diğerlerinin sayısı 5, "iyi" değerlerinin sayısı da 5 tir. O halde bu durum $|C_1| = 5$ ve $|C_2| = 5$ şeklinde ifade edilir. Küme içinde 10 eleman olduğuna göre $p_1 = 5/10 = 1/2$ ve $p_2 = 5/10 = 1/2$ olasılıkları hesaplanır. O halde RİSK kümesinin içerdiği "kötü" ve "iyi" değerler için olasılık dağılımını ifade edelim:

$$P_{RİSK} = \left(\frac{1}{2}, \frac{1}{2}\right)$$

$H(RİSK) = -\sum_{i=1}^{n} p_i \log_2(p_i)$ olduğuna göre, RİSK niteliğinin entropisi şu şekilde hesaplanır:

$$H(RİSK) = -\left(\frac{1}{2}\log_2 1/2 + \frac{1}{2}\log_2 1/2\right) = 1$$

Öncelikle BORÇ niteliğinin her bir değerinden kaç tane içerdiği belirlenir.

$|BORÇ_{YÜKSEK}| = 3$
$|BORÇ_{DÜŞÜK}| = 7$

BORÇ niteliğinin RİSK hedef niteliğindeki karşılıklarına baktığımızda, BORÇ niteliğinin yüksek niteliği karşısında RİSK niteliğinin 3 adet kötü değeri karşılık olmaktadır. Buna karşılık, BORÇ niteliğinin 7 adet düşük değeri için RİSK üzerinde 5 adet iyi, 2 adet kötü değeri karşılık gelmektedir. O halde $H(BORÇ_{YÜKSEK})$ ve $H(BORÇ_{DÜŞÜK})$ entropilerini hesaplayalım:

$$H(BORÇ_{YÜKSEK}) = -\left(\frac{3}{3}\log_2 3/3 + \frac{3}{3}\log_2 3/3\right) = 0$$

$$H(BORÇ_{DÜŞÜK}) = -\left(\frac{5}{7}\log_2 5/7 + \frac{5}{7}\log_2 5/7\right) = 0{,}863$$

Burada BORÇ niteliğinin değerlerine göre bir dallanma yapmak istediğimizde bu işlemin bize kazancı ne olacaktır? Söz konusu kazancı hesaplamak için bu niteliğin değerlerine göre entropilerini hesaplamalıyız. Buna göre:

$$H(X,T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} H(T_i)$$

olduğuna göre,

$H(BORÇ, RİSK) = 3/10H(BORÇ_{YÜKSEK}) + 7/10\ H(BORÇ_{DÜŞÜK})$
elde edilir. Bu durumda,

$H(BORÇ, RİSK) = 3/10(0) + 7/10(0.863) = 0.64$

hesaplanır. Kazanç ölçütü Kazanç(X, T) = H(T) – H(X, T) olduğuna göre

KAZANÇ(BORÇ, RİSK) = 1 – 0.64 = 0,36
elde edilir.

Dallanma için niteliklerin seçilmesi için kazanç oranı

Karar ağacının oluşturulması esnasında "kazanç ölçütü" adı verilen bir değeri kullandık. Ancak uygulamada bu yöntemden daha iyi sonuçlar veren bir başka yöntem daha kullanılmaktadır. Bilgi bölünmesi(split information) adı verilen bu kavram 1993 yılında Quinlan tarafından ortaya atılmıştır.

T kümesi için X niteliğini değerini belirlemek için gerekli bilgi miktarının ne olacağını bulmak için bu yol bulunmuştur. Söz konusu bilgi miktarı $H(P_{X,\ T})$ biçiminde ifade edilebilir ve $P_{X,\ T}$ ifadesi de X değerinin olasılık dağılımını ifade ederse bunun hesabı şu şekilde yapılır [74]:

$$P_{X,T} = \left(\frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \ldots, \frac{|T_k|}{|T|} \right) \quad (4.6)$$

Burada $H(P_{X,T})$ miktarı T kümesindeki X niteliği için "bilgi bölünmesi" olarak ifade edilmektedir. Bu değerlerin hesaplanması şu şekildedir:

$$H(P_{X,T}) = H\left(\frac{|T_1|}{|T|}, \frac{|T_2|}{|T|}, \ldots, \frac{|T_k|}{|T|} \right) \quad (4.7)$$

Bunun yerine aşağıdaki ifadeyi de kullanabiliriz:

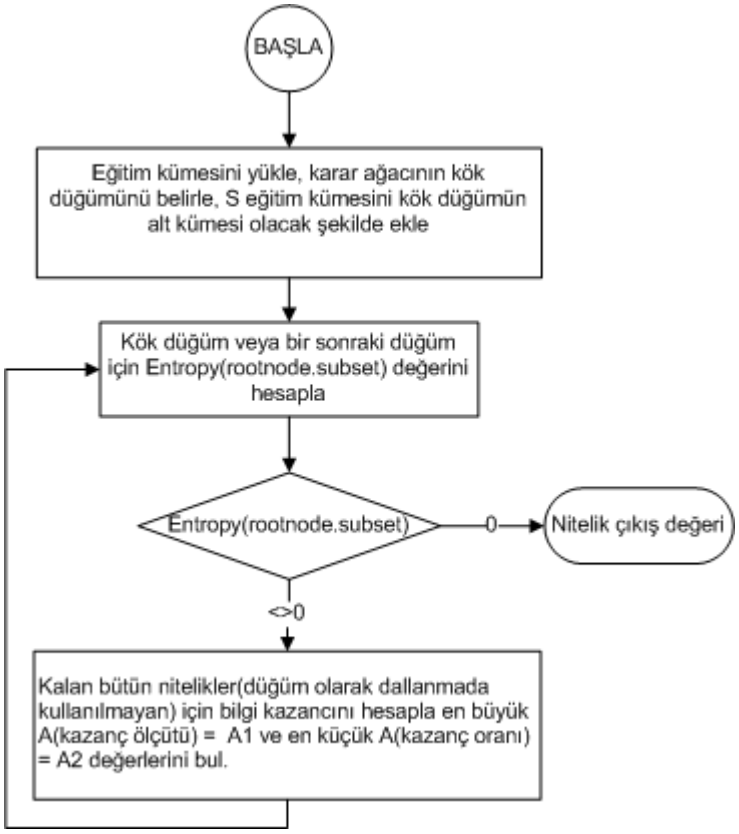
$$H(P_{X,T}) = -\sum_{i=1}^{k} \frac{T_i}{T} \log_2 \left(\frac{T_i}{T} \right) \quad (4.8)$$

Yukarıda elde edilen $H(P_{X,T})$ değeri ve kazanç ölçütü yardımıyla "kazanç oranı" hesaplanır:

$$\text{Kazanç Oranı}(X, T) = \text{Kazanç}(X, T) / H(P_{X,T}) \quad (4.9)$$

Uygulamada ID3 algoritması için önceki bölümlerde bahsedilen kazanç ölçütü ve kazanç oranı kriterleri kullanılabilmektedir. Ancak uygulamada, kazanç ölçütü kriterine göre daha iyi sonuç veren kazanç oranı kriteri kullanılmaktadır.

Şekil 4.4'te standart ID3 algoritmasının akış şeması gösterilmektedir.



Şekil 4.4. Standart ID3 algoritması akış şeması

4.3. C4.5 Algoritması

C4.5, ID3 algoritması gibi karar ağacı oluşturmayı sağlayan Quinlan'ın geliştirmiş olduğu algoritmalardan birisidir. C4.5, ID3 algoritmasının gelişmiş versiyonudur. C4.5 tarafından oluşturulan karar ağaçları sınıflandırma için kullanılabildiği için, C4.5 literatürde daha çok statiksel sınıflandırıcı olarak bilinmektedir [76].

4.3.1. Algoritmanın ID3' e göre artıları

C4.5 de aynı ID3 gibi belirli bir eğitim kümesini giriş olarak alıp buna göre bir karar ağacını entropi kavramını temel alarak oluşturur. C4.5 algoritması ID3 algoritmasının bütün özelliklerini kendisine miras alarak oluşturulmuş bir algoritmadır [74]. C4.5 algoritmasının geliştirilmesinin en önemli sebebi ID3

algoritmasının bazı eksiklikleri ve sorunları olmasıdır. C4.5 sisteminin ID3'e ek olarak getirdiği yaklaşımlar aşağıdaki gibi sıralanabilir:

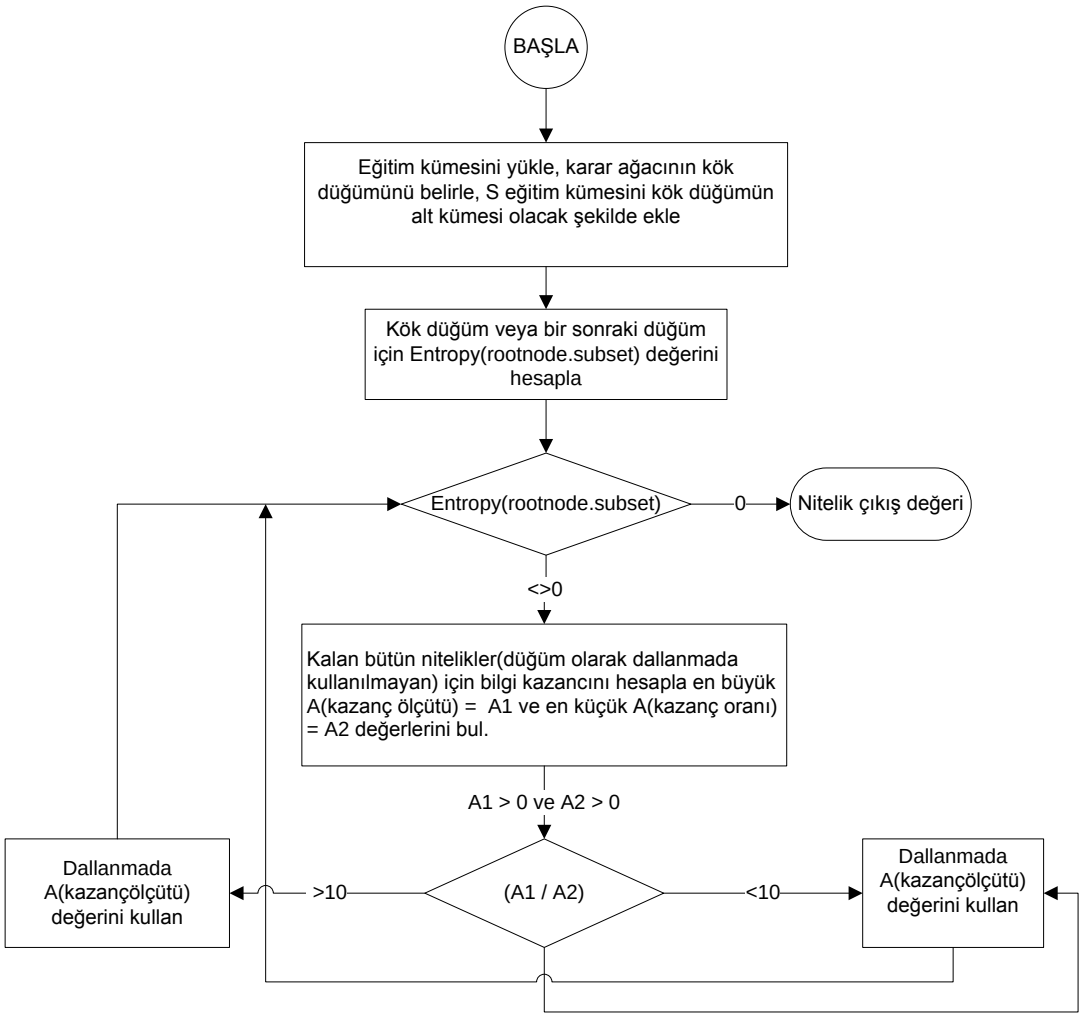
- Sürekli ve sayısal değerli niteliklere sahip verilerle başarılı bir şekilde çalışmayı sağlamak
- Nitelik değerleri eksik olan eğitim verileriyle başarılı bir şekilde çalışmayı sağlamak
- Karar ağacı oluşturulduktan sonra ağacın budanmasını sağlamak

Bahsedilen bu gelişmeler sonucunda C4.5'in özellikle sayısal veriler ve kayıp veriler üzerinde ID3 algoritmasına göre çok daha başarılı sonuçlar verdiği görülmektedir. Bu çalışmada kullanılan KDD Cup'99 verileriyle C4.5'in başarımı da deneysel sonuçlar bölümünde Çizelge 5.8 ve Çizelge 5.9'da sunulmuştur.

4.4. Önerilen Yöntem

Kazanç oranı ve kazanç ölçütü kriterlerinin her birinin daha iyi başarım gösterdiği entropi değerleri vardır. Kazanç ölçütü kriteri referans alındığında, dallanma yapılacak nitelik seçilirken kazanç ölçüt değeri en yüksek olan kök(ya da bir sonraki düğüm) düğüm olarak seçilir; kazanç oranı kriteri referans alındığında ise, kazanç oranı en düşük olan kök(ya da bir sonraki düğüm) düğüm olarak seçilir.

Karar ağacı oluşturulurken bu iki kriter de kullanılabilir. Standart ID3 algoritmasında kazanç oranı temel alınarak karar ağacı oluşturulmuştur. Önerilen yöntemin amacı bu iki yaklaşımın birbiriyle entegre edilmesini sağlamak ve standart ID3 algoritmasından daha iyi bir başarım elde etmektir. Önerilen yöntemin akış şeması Şekil 4.5'de gösterildiği gibidir.



Şekil 4.5. Önerilen algoritmanın akış diyagramı

Şekil 4.5'de görüldüğü gibi kazanç ölçütü ve kazanç oranı arasında bir katsayı belirlenmiştir ve bu katsayıya göre dallanılacak nitelik ya kazanç ölçütüne göre, ya da kazanç oranına göre seçilmektedir. Burada katsayı değeri 10 olarak gösterilmektedir. Bu katsayının belirlenmesi yapılan deneyler sonucunda belirlenmiştir. Deneysel sonuçlar bölümünde katsayı değerleri (5, 10, 15) için alınan sonuçlar Çizelge 5.3'te detaylı olarak sunulmuştur. Bu açıklamalardan sonra önerilen yöntemin sembolik olarak gösterimini aşağıdaki gibi gösterebiliriz.

X: Kazanç ölçütü

Y: Kazanç oranı

Z: Önerilen Yöntem Ölçütü

```
IF ( X < 0) Z = Y
IF ( X > 0 VE Y > 0)
{
      IF ( X / Y > 10)
            Z = X
}
ELSE
      Z = Y;
```

Sembolik olarak ifade edilen önerilen yöntemden şu çıkarımlar yapılabilir:

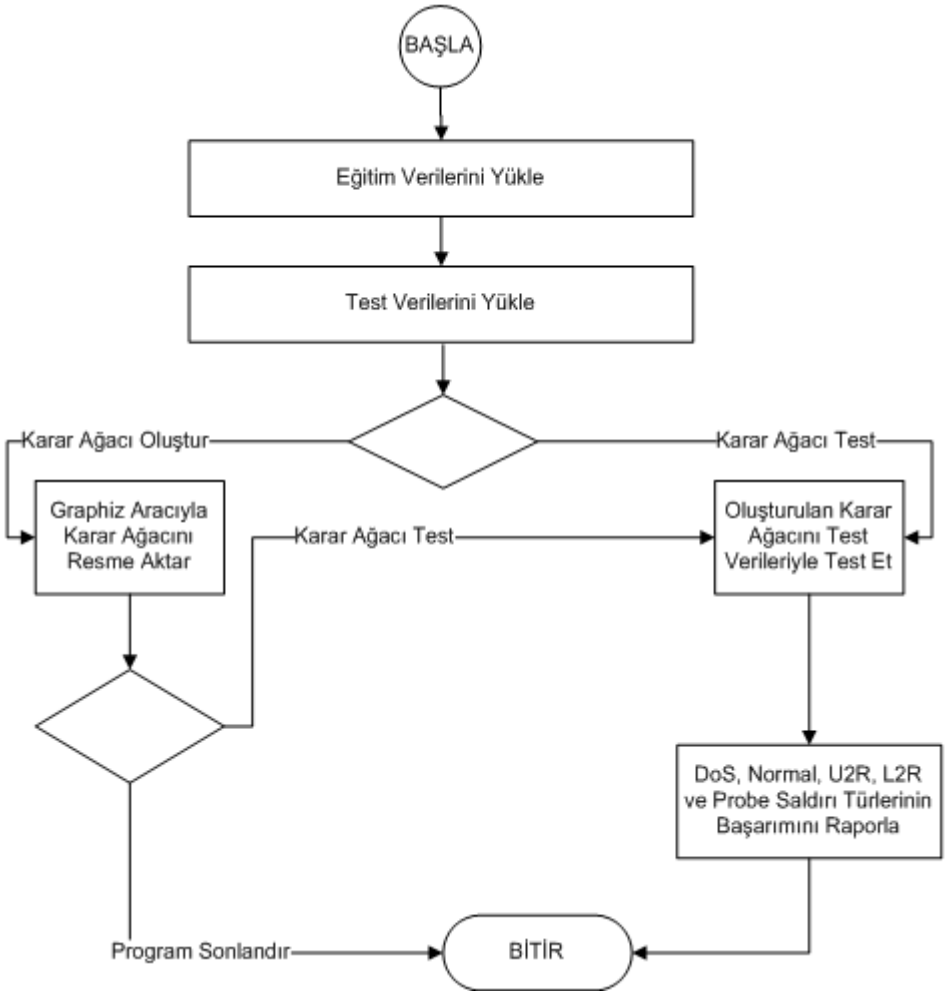
- Kazanç ölçütü değeri < 0 olduğu durumda dallanılacak nitelik kazanç oranı kriterine göre yapılır.
- Kazanç ölçütü ve kazanç oranının her ikisinin değerleri > 0 olduğu durumda; kazanç ölçütü değeri, kazanç oranı değerinin 10 katından daha büyükse dallanılacak nitelik kazanç ölçütü kriterine göre yapılır.
- Bunların dışında kalan durumlarda dallanılacak nitelik kazanç oranı kriterine göre yapılır.

4.5. Geliştirilen Saldırı Tespit Algoritması

Bu tez kapsamında geliştirilen yazılım aracılığıyla aşağıdaki fonksiyonlar gerçeklenmektedir:

- Eğitim verilerinin yüklenmesi
- Test verilerinin yüklenmesi
- Eğitim veri setiyle karar ağacının standart ID3 algoritmasına göre oluşturulması ve resme aktarılması
- Eğitim veri setiyle karar ağacının önerilen yönteme göre oluşturulması ve resme aktarılması

- Standart ID3 ve önerilen yönteme göre oluşturulan karar ağaçlarının test veri tabanıyla test edilmesi
- Standart ID3 ve önerilen yönteme göre başarım oranlarının raporlanması
- Standart ID3 ve önerilen yönteme göre saldırı tiplerinin başarım oranlarının raporlanması

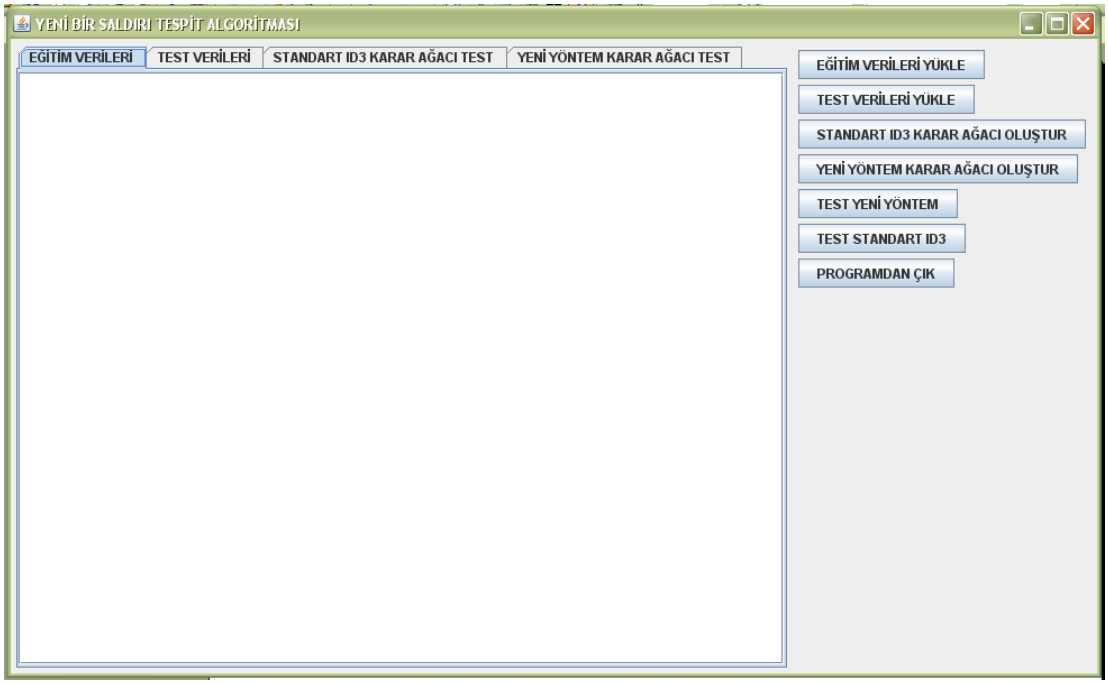


Şekil 4.6. Programın çalışma aşamaları

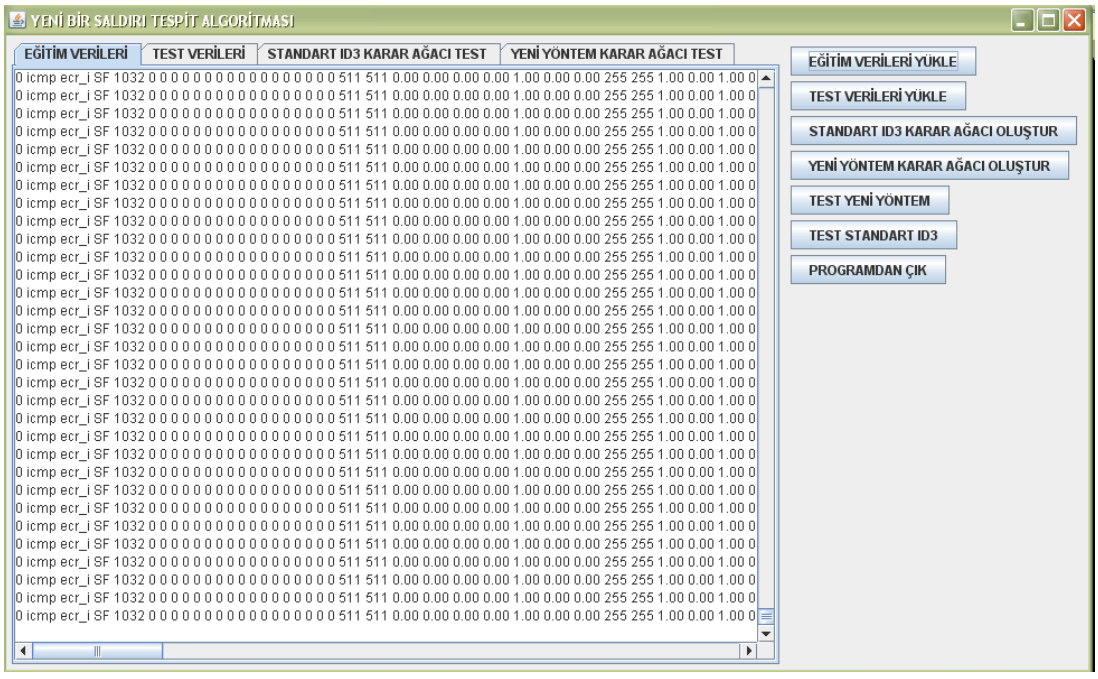
4.5.1. Programın bileşenleri

Şekil 4.7'de programın ilk çalışma ekran, Şekil 4.8'de karar ağacının oluşturulması için gerekli olan eğitim verilerinin yüklendiği ekran, Şekil 4.9'da karar ağacının başarımının test edilmesi için gerekli olan test verilerinin yüklendiği ekran, Şekil

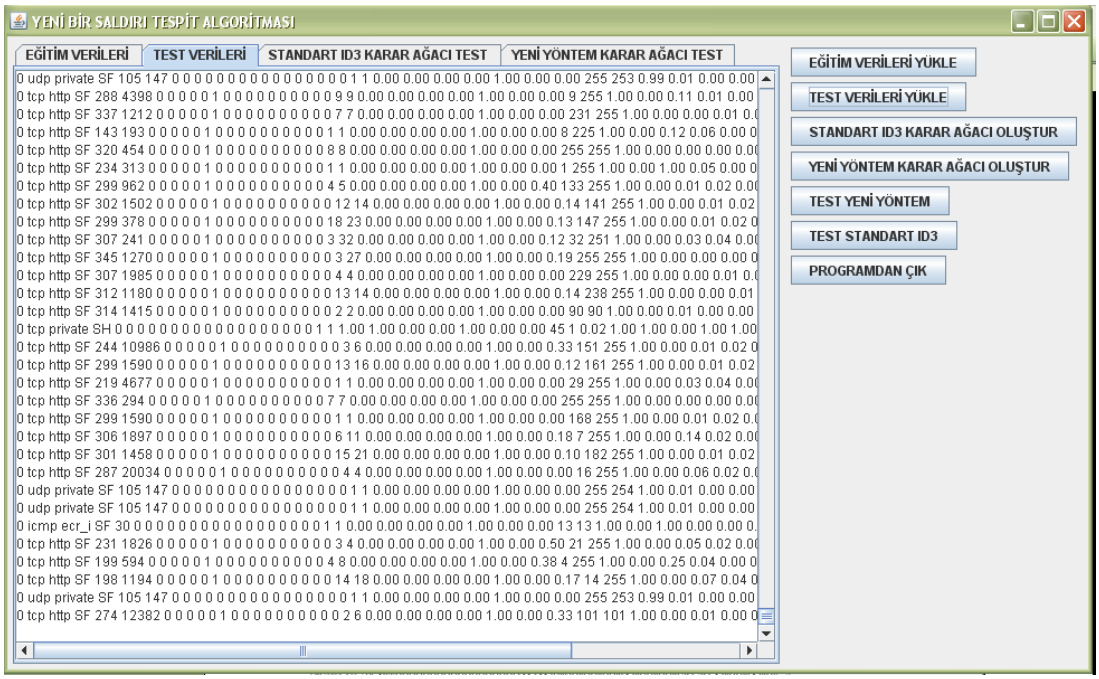
4.10'da önerilen yönteme göre karar ağacının test edildiği ve hangi saldırı tipi için başarılı olduğunu gösteren ekran görüntüleri sunulmuştur.



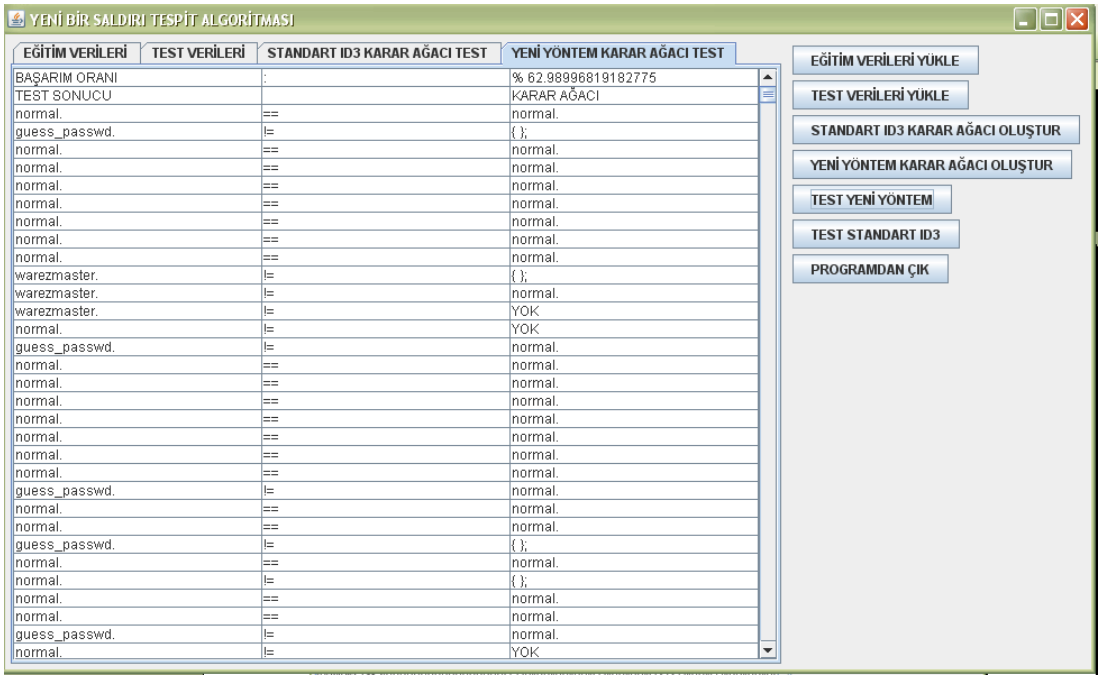
Şekil 4.7. Programın ilk çalışma ekranı



Şekil 4.8. Eğitim verilerinin yüklendiği ekran



Şekil 4.9. Test verilerinin yüklendiği ekran



Şekil 4.10. Önerilen yöntemin test edildikten sonraki ekran görüntüsü

Eğitim verileri yükle: Bu buton; karar ağacının oluşması için örnek eğitim kümemizi bilgisayarımın bulunduğu konumdan seçmemizi ve text alanında görüntülememizi sağlamaktadır. Bu işlevi gerçeklemek üzere `readFileIntoTextArea()` fonksiyonu çağrılmıştır. Şekil 4.8'de eğitim verilerinin içeriği text alanında gösterilmiştir.

Test verileri yükle: Karar ağacının test edilmesi için test veri kümemizi bilgisayarımın bulunduğu konumdan seçmemizi ve text alanında görüntülememizi sağlayan bir butondur. Bu işlevi gerçeklemek üzere `readFileIntoTextArea()` fonksiyonu çağrılmıştır. Şekil 4.9'da test verilerinin içeriği text alanında gösterilmiştir.

Standart Id3 karar ağacı oluştur / Yeni yöntem karar ağacı oluştur: Bu buton, yüklenmiş olan eğitim verileri kullanılarak karar ağacının oluşturulması sağlamaktadır. Karar ağacının oluşturulması için rekürsif fonksiyon olan `decomposeNode()` fonksiyonu çağrılmakta ve karar ağacı oluşturulmaktadır. d`ecomposeNode()` fonksiyonu ID3 algoritmasının temelini oluşturur. Bu fonksiyon aracılığıyla ağacın hangi niteliğe göre dallanacağı, niteliklerin entropilerinin hesaplanması ve kazanç ölçütünün hesaplanması sağlanarak kökte vektör olarak tutulan nitelik ve değerleri çocuk düğümlere bölünerek yaprak düğüme kadar gidecek ağacın oluşturulması sağlanır. Ağaç oluşturulduktan sonra `printTreeToDotCode()` fonksiyonuyla oluşturulan ağaç resme aktarılmaktadır. Bu fonksiyon Graphviz aracı ile Java arasında ara bir koddur. Bu kodun fonksiyonu, oluşturulan ağacın Graphviz aracına giriş olarak verilmesini sağlamaktadır. Graphviz aracı açık kaynak kodlu bir yazılım olmakla beraber özellikle ağaç – düğüm arasındaki bağlantıları göstermekte çok başarılı bir yazılım olduğu bilinmektedir. Yapılan bu uygulamada oluşan karar ağacının görsel olarak ifade edilebilmesi için bu aracın kullanılması oldukça önemlidir.

Test yeni yöntem / Test standart Id3: Bu butonun fonksiyonu eğitim verileriyle oluşturulan karar ağacında belirlenen saldırı tipiyle, test verilerindeki son nitelik olan

saldırı tipini belirleyen niteliğin değerlerinin kıyaslanması ve bunun sonucunda başarım oranının raporlanmasıdır. Burada `exploreDecisionTree()` ve `testDecisionTree()` fonksiyonları bu işlevin gerçeklenmesini sağlar. `ExploreDecisionTree()` fonksiyonu rekürsif bir fonksiyondur. Bu fonksiyon oluşturulan karar ağacındaki kurallara göre saldırı verisinin saldırı tipinin değerinin ne olduğunu bulur. `TestDecisionTree()` fonksiyonu ise saldırı verisindeki saldırı tipinin değeriyle `exploreDecisionTree()` fonksiyonundan geri dönen değerle karşılaştırma yapar ve değerlerin birbirine eşit olması durumunda *eşitSayisi* adında bir değişkenin arttırılması yoluyla başarım oranını elde eder. Şekil 4.11.'de bu durum gösterilmektedir.

Programdan çık: Bu buton programın sonlanması sağlar.

Geliştirilen uygulamada kullanılan önemli veri yapıları, değişkenler ve fonksiyonların açıklamaları EK-1' de sunulmuştur.

5. DENEYSEL SONUÇLAR

Bu tez kapsamında karar ağacı oluşturma yaklaşımlarından ID3 ele alınmış ve standart ID3 ile oluşturulan karar ağacının başarımıyla bu tezde sunulan yaklaşımla oluşturulan karar ağacının başarımlarının karşılaştırılması yapılmıştır.

Uygulama Eclipse geliştirme ortamı kullanılarak Java programlama dili ile geliştirilmiştir. Deneyler, 2.0 GHz, 4MB L2 önbellek, çift işlemcili, 160 GB disk ve 2.0 GB Ram teknik özelliklerine sahip bir diz üstü bilgisayar üzerinde yapılmıştır.

Standart ID3, karar ağacının ayrıştırılmasında, kazanç oranı kriterini temel almaktadır. Bu tezde sunulan yaklaşım, kazanım ve kazanç oranı kriterlerinin her ikisinin de en iyi sonuçlar verdiği koşulları göz önünde bulundurur ve bunların birbirine entegre edilmesiyle başarım oranını artırmaktadır. Bu çalışmada bu iki yaklaşımla karar ağacı oluşturma, karar ağacının test edilmesi ve bu yaklaşımların başarım oranlarının karşılaştırılmasını gerçeklemeye yönelik Java tabanlı bir program yazılmıştır. Programın çalıştırılıp test edilmesinden sonra alınan sonuçlar iki yaklaşımın hangi durumlarda daha iyi sonuçlar verdiklerini göstermektedir. Ayrıca Java ortamında oluşturulan ağacın açık kaynak kod olan Graphviz aracıyla karar ağacının resme aktarılması da yapılan bu uygulama ile gerçeklenmiştir.

Aşağıdaki bölümlerde sistemin eğitilmesi ve test edilmesi sırasında kullanılan KDD Cup'99 veri kümesinde kullanılan saldırı tipleri ve saldırı grupları hakkında bilgi verilip, eğitim ve test bölümüne geçilmiştir. Bu bölümde eğitim ve test verisi içinde kullanılan saldırı tipleri, saldırı grupları, bunların oranları ve bu veri kümeleri için iki yaklaşımdan da alınan sonuçlar hem her saldırı tipi için başarım sayıları hem de başarım yüzdeleri bakımından detaylı bir şekilde tablolar halinde sunulmuştur. Son olarak iki yöntemin birbirine olan üstünlüklerini daha açık olarak göstermek için tablolardan alınan sonuçlar resimlere aktarılmış ve bu resimlerin yorumları yapılmıştır. Ayrıca, bu bölümün sonunda C4.5 algoritmasının da KDD Cup'99 veri kümeleriyle test edilip başarım değerleri sunulmuştur [77].

5.1. KDD Cup'99 Veri Kümesi

Bu çalışmada kullanılan veri kümesi KDD Cup veri kümesinden alınmıştır. KDD Cup'99 veri kümesi 3. Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları yarışmasında ilk defa kullanılmıştır. Bu veri kümesi 1999 yılında MIT Üniversitesi tarafından DARPA verilerinden seçilerek ve düzenlenerek oluşturulmuştur. Bunun amacı saldırı tespit algoritmasının etkinliğinin değerlendirilmesidir [85]. Bundan dolayı bu çalışmada da bu veri kümesi kullanılmıştır.

Bu veri kümesinde, yaklaşık 4.940.000 çeşit eğitim verisi, 3.110.291 çeşit test verisi ve toplam 47 çeşit network bağlantı tipi(sürekli ve ayrık olmak üzere) bulunmaktadır. 23 çeşit saldırı tipi eğitim kümesinde, 37 çeşit saldırı tipi de test veri kümesinde bulunmaktadır. 14 çeşit saldırı tipi test veri kümesinde daha fazla olduğu için bilinmeyen saldırı tiplerinin tespit edilmesi için kullanılabilir. Test veri kümesinde kullanılan saldırı tipleri aşağıdaki gibi gruplara ayrılabilir [77]:

Çizelge 5.1 saldırı örüntülerini ve tiplerini listeler. Çizelge 5.2 veri miktarlarına göre eğitim kümesinde saldırı tiplerinin veri miktarlarını ve bulunma yüzdelerini listelemektedir [77].

Çizelge 5.1. KDD Cup'99 saldırı örüntüleri ve sınıflandırılması

Saldırı Tipi	**Saldırı Örüntüsü**
Probe	Ipsweep, nmap, portsweep, satan, mscan, Saint
DoS	back, land, neptune, pod, smurf, teardrop, apache2, mailbomb, processtable, udpstorm
U2R	Buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, httptunnel, xterm
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, snmpgetattack, named, xlook, xsnoop, snmpguess, worm, sendmail

Çizelge 5.2. Eğitim verilerinde kullanılan verilerin yüzdelik miktarları

Veri Tipi	Miktarı	Yüzdesi(%)
Normal	97.277	19.69
Probe	4107	0.83
DoS	391.458	79.24
U2R	52	0.01
R2L	1126	0.22

5.2. Eğitim ve Test

Bu çalışma değişik şartlar altında standart ID3 algoritmasıyla sunulan yeni yaklaşımın etkinliğinin karşılaştırılmasını gerçeklemektedir. KDD Cup'99 veri kümesi çok büyük, çeşitli ve düzgün dağılmamış bir veri kümesi olduğu için bu çalışmada bu veri kümesinin eğitim ve test verilerinin %10'u(10% kddcup.data_10_percent.gz) kullanılarak gerçeklenmiştir. Veritabanındaki dağılımına bağlı olarak hem eğitim kümesinden hem de test kümesinden %10luk, %20'lik, %30'luk,…,%90'lık veriler örneklenip [77] verilerin başarımlarının karşılaştırılması Çizelge 5.4'de gösterilmiştir .

Çizelge 5.3'te önerilen yöntemde belirlemek istediğimiz katsayı değerleri (K = 5, 10, 15) için almış olduğumuz sonuçlar sunulmuştur. Buna göre önerilen yöntemde en iyi sonuçları K = 10'da aldığımız için önerilen yöntemde katsayı değerimiz (K = 10) olarak kabul edilmiştir.

Çizelge 5.3. Önerilen yöntemin K = 5, 10, 15 değerleri için elde edilen başarım oranları ve bu oranların ortalamaları

%Normal Veri	%Önerilen Yöntem (K = 5)	%Önerilen Yöntem (K = 10)	%Önerilen Yöntem (K = 15)
10	66,9244	65,3046	63,6017
20	71,0381	34,2199	34,2247
30	55,4243	39,4914	39,4881
40	45,2956	42,3596	42,2947
50	41,4903	59,9277	43,5084
60	41,536	56,8858	52,9238
70	48,1042	60,7396	59,447
80	50,5254	60,2957	52,977
90	60,4387	65,3151	66,3395
Ortalama	53,4196	53,8377	50,5339

Çizelge 5.4. Standart ID3 yaklaşımıyla sunulan yeni yaklaşımın süre ve başarım olarak performansını gösteren değerler tablosu

%Normal Veri	%Standart ID3	Süre(sn)	%Önerilen Yöntem	Süre(sn)
10	48,1331	38	65,3046	93
20	56,1939	135	34,2199	146
30	39,4897	353	39,4914	379
40	42,2630	388	42,3596	436
50	43,4741	463	59,9277	930
60	52,8953	565	56,8858	1137
70	59,3869	764	60,7396	1175
80	52,9287	869	60,2957	1342
90	66,2966	936	65,3151	1412
Ortalama	51,2290	501,2222	53,8377	783

Çizelge 5.5. Standart ID3 yaklaşımına göre saldırı tiplerinin belirtilen veri miktarları için test verisine karşılık gelen doğruluk değerleri

% Normal Veri	Probe	Normal	U2R	DoS	R2L
10	277 1457	9457 9782	0 18	98 6626	4 2552
20	1231 1492	11096 11394	0 66	10636 25124	4 2795
30	1455 1865	20694 21244	0 191	1904 27761	157 10246
40	1724 2255	30521 31443	0 193	2145 37606	157 10246
50	1724 2255	30752 31674	0 193	11788 57810	157 10246
60	1724 2255	30752 31674	0 193	32224 78246	157 10246
70	1885 2276	37519 38604	0 197	45392 91414	157 10559
80	1895 2280	41215 42555	0 198	43264 106401	157 12052
90	1895 2280	41215 42555	0 198	78667 126837	157 12052

Çizelge 5.6. Önerilen yaklaşıma göre saldırı tiplerinin belirtilen veri miktarları için test verisine karşılık gelen doğruluk değerleri

% Normal Veri	Probe	Normal	U2R	DoS	R2L
10	21 1457	7083 9782	0 18	6236 6626	5 2552
20	1230 1492	11094 11394	0 66	1658 25124	4 2795
30	1458 1865	20697 21244	0 191	1899 27761	157 10246
40	1727 2255	30569 31443	0 193	2173 37606	157 10246
50	794 2255	27707 31674	0 193	32594 57810	138 10246
60	794 2255	27707 31674	0 193	41111 78246	138 10246
70	906 2276	33782 38604	0 197	52062 91414	138 10559
80	963 2280	34190 42555	0 198	63284 106401	138 12052
90	959 2280	34194 42555	0 198	84838 126837	138 12052

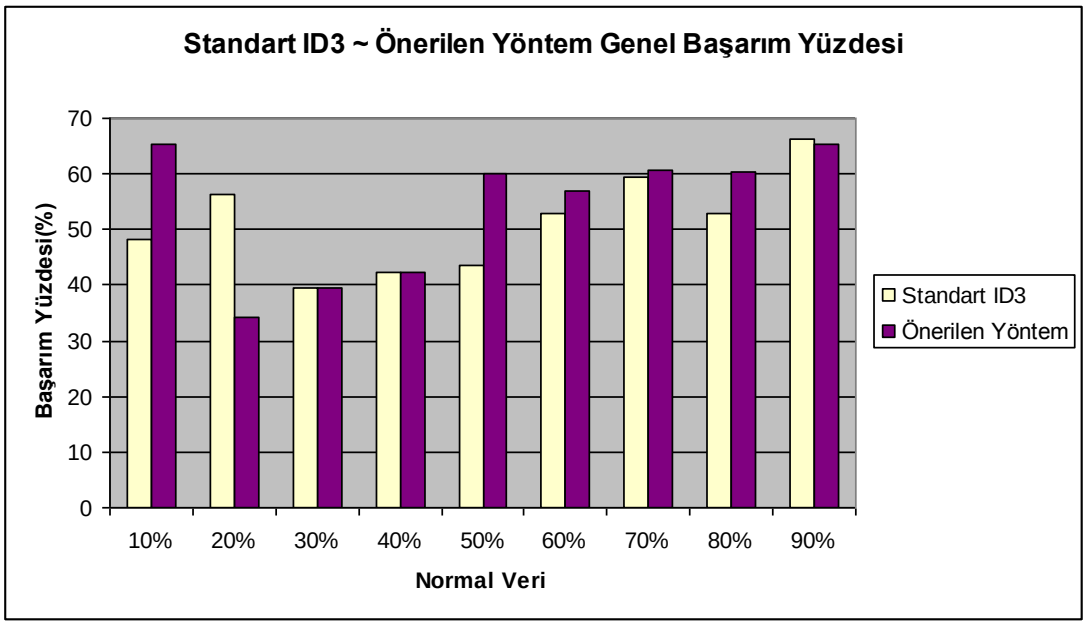
Çizelge 5.5 ve Çizelge 5.6'da ilk satırları algoritmaların doğru tespit ettiği saldırı tipi sayısını gösterirken, ikinci satırları ise test verilerinde o saldırı tipinden kaç tane olduğunu göstermektedir. Çizelge 5.7'de ise her iki algoritma için probe, normal, U2R, DoS ve R2L'in başarım yüzdelerini göstermektedir.

Çizelge 5.7. Önerilen yaklaşıma ve standart ID3'e göre saldırı tiplerinin belirtilen veri miktarları için başarım yüzdeleri

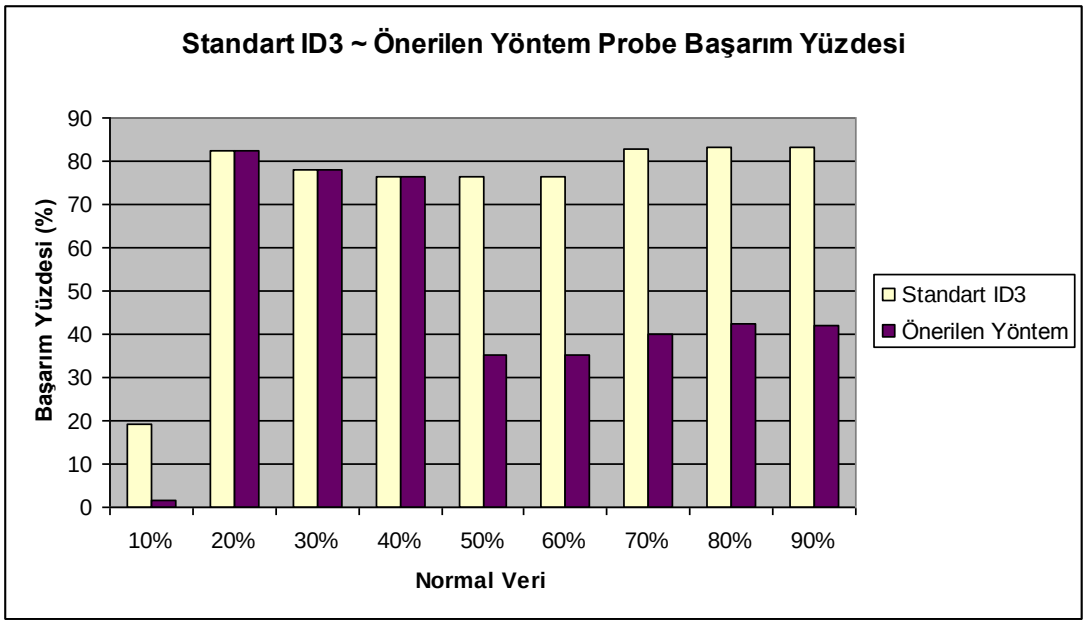
% Normal Veri	Yaklaşımlar	%Probe	%Normal	%U2R	%DoS	%R2L
10	Standart ID3	19,0117	96,6776	0	1,4790	0,1567
	Önerilen Yöntem	1,4413	72,4085	0	94,1141	0,1959
20	Standart ID3	82,5067	97,3846	0	42,3340	0,1431
	Önerilen Yöntem	82,4397	97,367	0	6,5993	0,1431
30	Standart ID3	78,0161	97,4110	0	6,8585	1,5323
	Önerilen Yöntem	78,1769	97,4252	0	6,8405	1,5323
40	Standart ID3	76,4523	97,677	0	5,7039	1,5323
	Önerilen Yöntem	76,5854	97,2204	0	5,7783	1,5323
50	Standart ID3	76,4523	97,0891	0	20,3909	1,5323
	Önerilen Yöntem	35,2106	87,4755	0	56,3812	1,3469
60	Standart ID3	76,4523	97,0891	0	41,1829	1,5323
	Önerilen Yöntem	35,2106	87,4755	0	52,5407	1,3469
70	Standart ID3	82,8207	97,1894	0	49,6554	1,4869
	Önerilen Yöntem	39,8067	87,5091	0	56,9519	1,3069
80	Standart ID3	83,1140	96,8511	0	40,6613	1,3027
	Önerilen Yöntem	42,2368	80,3431	0	59,4769	1,1450
90	Standart ID3	83,1140	96,8511	0	62,0221	1,3027
	Önerilen Yöntem	42,0614	80,3525	0	66,8874	1,1450

Aşağıdaki şekiller yardımıyla standart ID3 yaklaşımıyla önerilen yaklaşımın eğitim verisiyle eğitilip test verisiyle test edilip genel başarım sonuçlarının karşılaştırılması ve normal, probe, R2L, U2R ve DoS saldırı gruplarının bu iki yaklaşıma göre başarımlarının hangi saldırı tipi için hangi yaklaşımın daha başarılı olduğu daha iyi anlaşılmaktadır. Hem standart ID3 yaklaşımında hem de önerilen yaklaşımda U2R saldırı grubu olan "buffer_overflow, perl, ps, xterm, loadmodule, rootkit, sqlattack, httptunnel" saldırı tipleri tanınamadığı için aşağıdaki şekillerde gösterilmemiştir. Bu durum Çizelge 5.5, Çizelge 5.6 ve Çizelge 5.7'de ayrıntılı olarak görülmektedir. Şekil 5.1 önerilen yöntemle, standart ID3 yönteminin genel başarım oranını, Şekil 5.2 önerilen yöntemle standart ID3 yaklaşımının Probe saldırı grubu için başarım

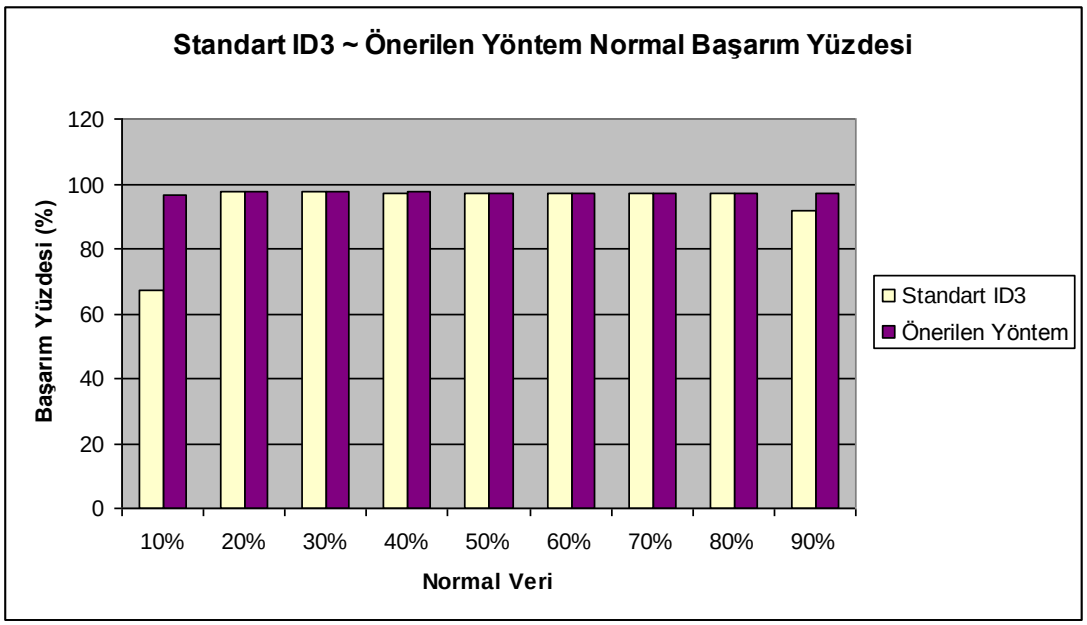
oranını, Şekil 5.3 önerilen yöntemle standart ID3 yaklaşımının Normal saldırı grubu için başarım oranını, Şekil 5.4 önerilen yöntemle standart ID3 yaklaşımının R2L saldırı grubu için başarım oranını, Şekil 5.5 önerilen yöntemle standart ID3 yaklaşımının DoS saldırı grubu için başarım oranını ve Şekil 5.6 önerilen yöntemle ve standart ID3 yaklaşımıyla oluşturulan karar ağacının oluşturulma sürelerini göstermektedir.



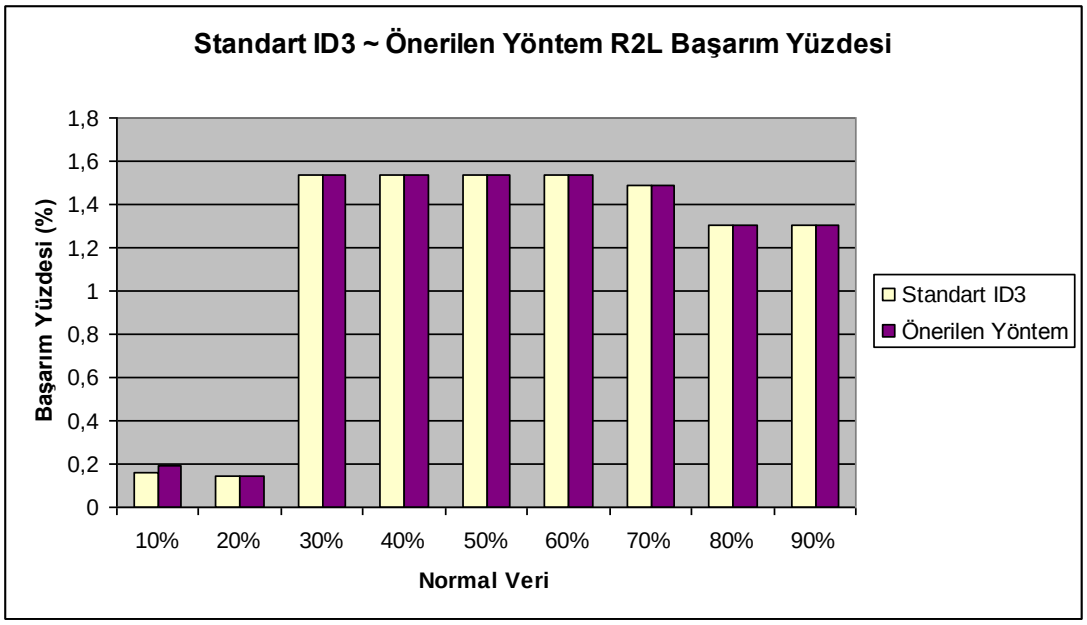
Şekil 5.1. Önerilen yöntemle standart ID3 yaklaşımının genel başarımı



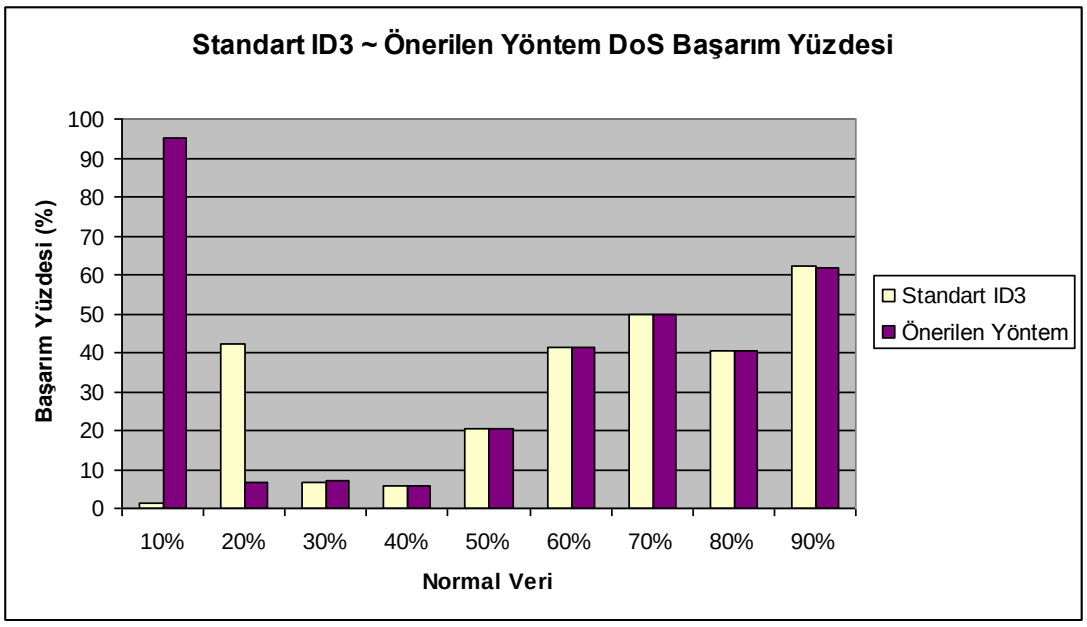
Şekil 5.2. Önerilen yöntemle standart ID3 yaklaşımının Probe saldırı grubu için başarımı



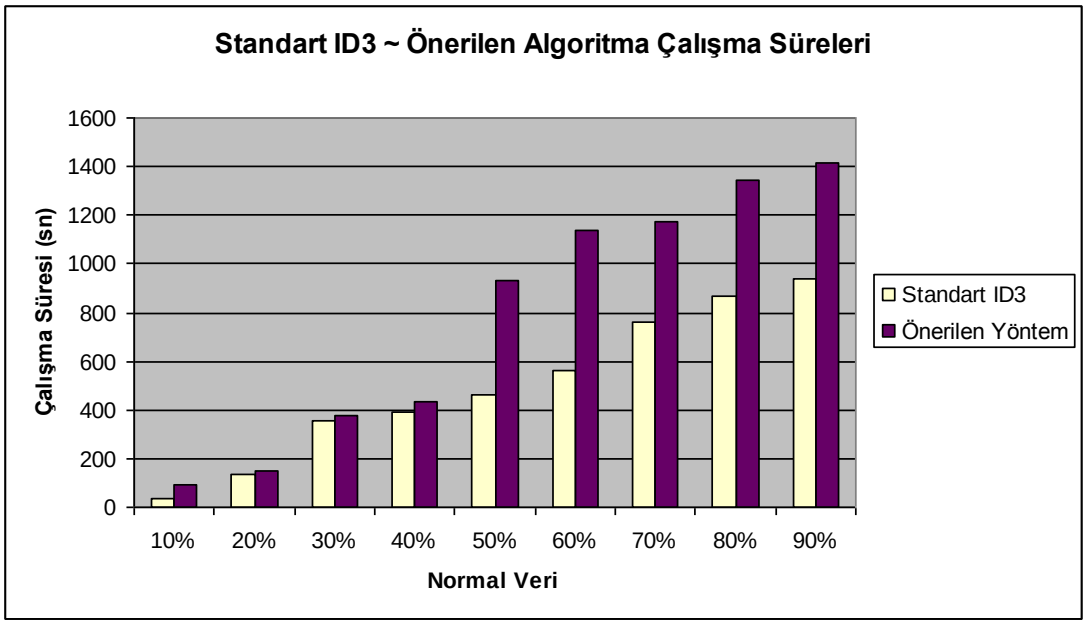
Şekil 5.3. Önerilen yöntemle standart ID3 yaklaşımının Normal saldırı grubu için başarımı



Şekil 5.4. Önerilen yöntemle standart ID3 yaklaşımının R2L saldırı grubu için başarımı



Şekil 5.5. Önerilen yöntemle standart ID3 yaklaşımının DoS saldırı grubu için başarımı



Şekil 5.6. Önerilen yöntemle standart ID3 algoritmalarının çalışma süreleri

- DoS saldırısı için: Önerilen yöntemin bu saldırı grubu için oldukça başarılı olduğunu söyleyebiliriz. Şekil 5.6'dan görüldüğü gibi sadece veri miktarının %10 - %20 arasında olması durumunda standart ID3 algoritmasının daha iyi sonuç verdiği söylenebilir, ancak diğer durumlarda önerilen yöntemin başarımı çok daha iyidir.
- U2R saldırısı için: Çizelge 5.3, Çizelge 5.4, Çizelge 5.5'ten de açıkça görüldüğü gibi ne standart ID3 ne de önerilen yöntem U2R saldırı grubunu tanıyamamıştır. Her iki yöntemde de bu saldırı tipinin hiç tanınamaması ID3 algoritmasının bir eksikliğini göstermektedir.
- R2L saldırısı için: Şekil 5.4'den görüldüğü gibi genelde yaklaşık olarak başarım oranları her iki yöntem için de aynıdır. Özellikle veri miktarının %40'ı aştığı durumlarda standart ID3 daha başarılıdır.
- Başarım ortalama değerlerine bakıldığında, önerilen yöntemin daha iyi bir başarım sergilediği Şekil 5.1'den açıkça görülmektedir.
- Normal veriler için, Şekil 5.3'den görüldüğü gibi yaklaşık iki yaklaşım da aynı değerlere sahiptirler. Sadece, %10'luk veri miktarı için standart ID3 başarım olarak daha iyi olduğu görünmektedir. Diğer saldırı tiplerine

oranlayacak olursak, normal verilerin sınıflandırılması her iki yaklaşım için de en başarılı olduğu grup olduğu açıkça görünmektedir..

- Şekil 5.6'dan görüldüğü gibi standart ID3 algoritması daha hızlı çalışmaktadır. Çünkü önerilen yöntemde hem kazanç oranı hem de kazanç ölçütü değerleri bulunarak bir karşılaştırma yapılır ve hangi kriter önerilen yöntem için uygunsa o yaklaşım seçilmektedir. Yapılan bu işlemlerden dolayı çalışma sürelerindeki bu farkın olması normaldir.

5.3. C4.5'in KDD Cup'99 Veri Kümesiyle Testi

Bölüm 4.3.1'de bahsedildiği gibi ID3 algoritması daha çok sembolik verilere cevap verebilmekte, sayısal verilerle ve kayıp verilerle çok başarılı sonuçlar vermemektedir. Bahsi geçen sorunların giderilmesi için Quinlan tarafından, aynı ID3 gibi entropi tabanlı bir algoritma olan C4.5 geliştirilmiştir.

Çizelge 5.8 ve Çizelge 5.9, KDD Cup'99 veri kümesi kullanılarak C4.5 algoritmasının sırasıyla genel başarım yüzdesi ve saldırı tipini tanıma yüzdelerini göstermektedir.

Çizelge 5.8. C4.5 algoritmasının belirtilen veri miktarları için genel başarım yüzdesi [77]

%Normal Veri	%C4.5
10	73,79
20	70,90
30	74,09
40	66,47
50	70,14
60	69,15
70	68,70
80	65,30
90	65,00
Ortalama	70,62

Çizelge 5.9. C4.5 için belirli veri miktarları için saldırı tiplerinin tanınma yüzdesi [77]

%Normal Veri	%Probe	%U2R	%DoS	%R2L
10	90,86	58,63	62,53	20,15
20	80,35	47,82	62,87	19,15
30	93,94	51,38	63	16,88
40	83,63	53,93	61,50	17,45
50	83,75	44,74	60,95	15,70
60	84,42	50,51	71,51	14,44
70	84,46	27,59	66,67	17,33
80	88,85	44,42	54,55	12,97
90	78,79	57,92	58,30	13,70

Çizelge 5.8 ve Çizelge 5.9'dan görüldüğü üzere C4.5'in hem standart ID3, hem de önerilen yöntemden daha iyi bir başarım gösterdiği görülmektedir. ID3 algoritmasının takipçisi olan C4.5 algoritmasının daha başarılı olmasının en önemli sebepleri aşağıdaki gibi sıralanabilir:

- Sürekli ve sayısal değerli niteliklere sahip verilerle başarılı bir şekilde çalışabilmek
- Nitelik değerleri eksik olan eğitim verileriyle başarılı bir şekilde çalışabilmek
- Karar ağacı oluşturulduktan sonra ağacın budanması

6. SONUÇ VE ÖNERİLER

Tıp, sosyo-ekonomik, savunma sanayi vb. gibi birçok alanda kullanılan karar destek sistemleri için karar ağacı oluşturma algoritmaları çok kritik bir öneme sahiptir. Bu çalışmada, KDD Cup'99 veri kümeleri kullanılarak veri madenciliğinin sınıflandırma tekniklerinden biri olan karar ağaçlarıyla bilgisayar ağlarında yeni bir saldırı tespit algoritması geliştirilmiş ve bu algoritmanın nasıl geliştirildiği üzerinde durulmuştur. KDD Cup'99 veri kümesiyle standart ID3 ve yeni önerilen yöntemle iki farklı karar ağacı oluşturulup, bu ağaçların tespit ettiği saldırı tipiyle test verisindeki saldırı tipi karşılaştırılarak her iki yöntemin başarım oranının raporlanması sağlanmıştır. Bu karşılaştırmanın sonuçları raporlanmış ve yeni önerilen yöntemin şu an uygulamada kullanılan ID3 algoritmasından ortalama %2 daha iyi bir başarım gösterdiği görülmüştür.

Mevcut ID3'te sadece kazanç oranı kriteri temel alınarak karar ağacı oluşturulurken, önerilen yöntemde hem kazanç oranı hem de kazanç ölçütü kriterlerinin birbirine entegre edilmesiyle karar ağacı oluşturulmuştur. Karar ağacı oluşturulurken en önemli aşaması dallanma ya da bölümlemenin hangi niteliğe göre yapılacağıdır. Bu dallanma yapılırken bazı niteliklerde kazanç ölçütü daha iyi başarım gösterirken, bazı niteliklerde ise kazanç oranı daha iyi bir başarım göstermektedir. Önerilen yöntem her iki ölçütün daha başarılı olduğu niteliklerin seçilmesini sağlamaktadır. Bu seçimi yapmak için niteliklerin kazanç ölçütü ve kazanç oranı değerleri hesaplanır ve bu kriterlerden hangisinin seçileceğini belirlemek için bir kriter belirlenmiştir. Bu kriterin nasıl belirlendiği dördüncü bölümde ayrıntılı bir şekilde anlatılmıştır. Deneysel sonuçlardan da görüldüğü gibi önerilen yöntem, standart ID3 algoritmasından daha iyi bir başarım göstermiştir. Ayrıca, literatürde özellikle sayısal değerli veriler için daha başarılı olan ID3'ün gelişmiş versiyonu olarak da bilinen C4.5 algoritmasının başarımı da Çizelge 5.8 ve Çizelge 5.9'da sunulmuştur.

Bu çalışmada da başvurulan KDD Cup'99 veri kümesi yaygın olarak şimdiki saldırı tespit sistemlerinde kullanılmakta ama bu 1999'un verisi olduğu için ve ağ teknolojileri ve saldırı metotları sürekli değiştiğinden bu, şimdiki gerçek ağ

durumunu yansıtamayabilir. Bu yüzden gelecekte yapılacak çalışmalarda, günümüzdeki ağ durumunu daha doğru bir şekilde yansıtabilmek için daha yeni bilgiye sahip olunup, test edilip ve kıyaslama yapılması önerilmektedir.

Gerçekleştirilen çalışma sonucunda, veri madenciliğinin sınıflandırma yöntemlerinden biri olan ve karar destek sistemleri için de çok önemli bir noktada bulunan karar ağaçları hakkında geniş bir bilgi birikimi elde edilmiştir. Entropiye dayalı olarak karar ağacı oluşturan algoritmalardan C4.5 ve ID3 hakkında bilgi edinilmiştir.

Bu tez çalışmasında karşılaşılan güçlükler;

- Toplanan verilerin işlenmesini uzun sürmesi
- Güncel saldırı verilerinin ücretsiz olmaması
- Standart ID3 ve önerilen yönteme göre karar ağaçlarının oluşturulmasının ve test edilmesinin uzun sürmesi
- Algoritmalar öğrenen algoritmalar olduğu için büyük verilere ihtiyaç duyulması

şeklinde sıralanabilir.

KAYNAKLAR

1. Dayıoğlu, B., Özgit, A., "İnternet'de saldırı tespiti teknolojileri", ***İletişim Teknolojileri 1. Ulusal Sempozyumu ve Fuarı***, Ankara, 1-5 (2001).

2. Sağıroğlu, Ş., Alkan, M., "Her yönüyle elektronik imza (e-imza)", ***Grafiker Yayınları***, Ankara, 1-100 (2005).

3. Spafford, E., "The internet worm program: an analysis", ***Purdue Technical Report CSD-TR-823, Dept. of Comp. Sciences, Purdue University, West Lafayette,*** 1-41 ***(***1988).

4. Spafford, E., "The internet worm incident", ***Technical Report CSD-TR-933, Dept. of Comp. Sciences, Purdue University, West Lafayette,*** 1-19 (1989).

5. Eichin, M., Rochlis, J.A., "With microscope and tweezers: an analysis of the internet virus of november 1998", ***Proceedings of the 1989 IEEE Symposium on Research in Security and Privacy,*** 1-18 (1989).

6. Güven, E. N., "Zeki saldırı sistemlerinin incelenmesi, tasarımı ve gerçekleştirilmesi", Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü,*** 1-91 (2007).

7. İnternet: Teknotürk.org kaynağından bilişim, "Veri madenciliği ile saldırı tespiti", http://www.teknoturk.org/ (2009).

8. Mahoney, M.V., Chan, P.K., "An analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for network anomaly detection", ***Recent Advances in Intrusion Detection (RAID2003),*** Lecture Notes in Computer Science., Springer-Verlag, 2820, 220-237 (2003).

9. İnternet: Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Mühendisliği Bölümü, "Anormallik tespit teknikleri: mevcut çözümler ve en son teknolojik gelişimler", http://www.bilmuh.gyte.edu.tr/ (2009).

10. Yarımağan, Ü., "Veritabanı sistemleri", ***Akademi & Türkiye Bilişim Vakfı Yayını,*** Ankara, 1-362 (2000).

11. Aydoğan, F., "E-Ticarette veri madenciliği yaklaşımlarıyla müşteriye hizmet sunan akıllı modüllerin tasarımı ve gerçekleştirimi", Yüksek Lisans Tezi, ***Hacettepe Üniversitesi Fen Bilimleri Enstitüsü,*** 1-179 (2003).

12. Akpınar H., "Veri tabanlarında bilgi keşfi ve veri madenciligi", ***İ.Ü. İşletme Fakültesi Dergisi,*** 1-22 (2000).

13. Alpaydın, E., "Zeki veri madenciliği: ham veriden altın bilgiye", ***Bilişim 2000 Eğitim Semineri,*** İstanbul, 1-10 (2000).

14. Frawley, W. J., Piatetsky-Shapiro, G., Matheus, C. J., "Knowledge discovery databases: An overview, In Knowledge Discovery In Databases", ***American Association for Artificial Intelligence,*** Cambridge, MA: AAAI/M1T**,** 1-27 (1991).

15. Han, J., Chiang, J., Chee, S., Chen, J., Chen, Q., Cheng, S., Gong, W., Kamber, M., Koperski, K., Liu, G., Lu, Y., Stefanovic, N., Winstone, L., Xia, B., Zaiane, O. R., Zhang, S., Zhu, H., "DBMiner: A system for data mining in relational datahases and data warehouses", ***Proc. CASCON'97: Meeting of Minds,*** Toronto, Canada, 1-79 *(*1997).

16. Woolf, R. J., "Data mining using matlab", ***University of Faculty of Engineering & Surveying,*** 1-137 (2000).

17. Kaya, E., Bulun, M., Arslan, A., "Tıpta veri ambarları oluşturma ve veri madenciliği uygulamaları"***, Akademik Bilişim Konferansları,*** 1-11 (2003).

18. Gürcan, F., Köse, C., "Web içerik madenciliği ve konu sınıflandırılması", ***VI. İstatistik Günleri Sempozyumu***, 1-5 (2008).

19. Alataş, B., Akın, E., "Veri madenciliğinde yeni yaklaşımlar", ***Yöneylem Araştırması/Endüstri Mühendisliği - XXIV Ulusal Kongresi,*** 1-3 (2004).

20. İnternet: Değer Üretme Sanatı, "Veri madenciligi veya bilgi keşfi", http://www.bilgiyonetimi.org/ (2009).

21. Akgöbek, Ö., Çakır, F., "Veri madenciliğinde bir uzman sistem tasarımı", ***Akademik Bilişim Konferansları,*** 1-5 (2009).

22. Gregory, P., "Knowledge discovery in real databases: a report on the IJCAI- 89 Workshop", ***Al Magazine,*** Cilt: 11, Sayı:5, 68-70 (1989).

23. Berry, M. J. A., Linoff, G., "The art and science of customer relationship management**"**, U.S.A., ***Wiley Computer Publishing,*** 7-14 (2000).

24. Adrians, P., Zantinge, D., "Data mining**"**, England, ***Addison- Wesley,*** 5 (1997).

25. Özmen, Ş., "İş hayatı veri madenciliği ile istatistik uygulamalarını yeniden keşfediyor", ***V. Ulusal Ekonometri ve İstatistik Sempozyumu,*** 1-7 (2001).

26. Şimşek, U. T., "Veri madenciliği ve müşteri ilişkileri yönetiminde(Crm) bir uygulama", Doktora Tezi, ***İstanbul Üniversitesi Sosyal Bilimler Enstitüsü,*** 1-182 (2006).

27. İnternet: CRISP-DM - process model, "Cross Industry Standard Process for Data Mining", http://www.crisp-dm.org/Process/index.htm (2009).

28. Altıntaş, T., "Veri madenciliği metotlarından olan kümeleme algoritmalarının uygulamalı etkinlik analizi", Yüksek Lisans Tezi, ***Sakarya Üniversitesi Fen Bilimleri Enstitüsü,*** 1-51 (2006).

29. Akpınar H., "Veri tabanlarında bilgi keşfi ve veri madenciligi", ***İ.Ü. İşletme Fakültesi Dergisi***, Sayı:1, İstanbul, 1-22 (2000).

30. Moss, L. T., "Business intelligence roadmap: the complete project lifecycle for decision- support applications", Germany, ***Addison Wesley,*** 307-320 (2003).

31. Jiawei, H., Micheline, K., Data mining concepts and techniques, U.S.A., ***Morgan Kaufman Publishers,*** 106 (2001).

32. Özmen, Ş., "Veri madenciliği süreci", ***Veri Madenciliği ve Uygulama Alanları Konferansı,*** İstanbul, 1-5 (2003).

33. Jiawei, H., Micheline, K., "Data mining concepts and techniques", U.S.A., ***Morgan Kaufman Publishers,*** 106-108 (2001).

34. Berry, M. J. A., Linoff, G., "Data mining solutions**",** U.S.A., ***Wiley Computer Publishing***, 14-106 (2000).

35. Baykal, N., "Veri tabanı ve veri madenciliği", ***Tıp Bilişimi Güz Okulu***, 61-64 (2003).

36. Rygielski, C.**,** Wang, J., Yen, D. C., "Data mining techniques for customer relationship management", ***Technology in Society,*** Volume 24, Issue 4, 488-490 (2002).

37. İnternet: Introduction to Fraud Detection, "Introduction to fraud detection", http://www.dinkla.net/fraud/types.html (2009).

38. Özel, T., "Finans sektöründe istatistik ve veri madenciliği uygulamaları", ***SPSS sunumu,*** 1-8 (2003).

39. Chye, K. H., Gerry, C. K. L., "Data mining and customer relationship management in the banking industry", ***Singapore Management Review,*** Volume 24, Issue 2, 1-27 (2002).

40. İnternet: SPSS Türkiye, "Veri madenciliği uygulama alanları", http://www.spss.com.tr/veri.htm (2009).

41. Roddick, J., "Exploratory medical knowledge discovery: experiences and issues", ***ACM SIGKDD Explorations Newsletter,*** Volume 5, Issue 1, 94-99 (2003).

42. Breault, J., Goodall, C., Fos, P., "Data mining a diabetic data warehouse", ***Artificial Intelligence in Medicine***, 37-54 (2002).

43. Simoudis, E., "Reality check for data mining", ***In IEEE Expert: Intelligent Systems and Their Applications ,*** ll(5), 26-33 (1996).

44. Michalski, R. S. and Stepp, R. E., "Learning from observation: Conceptual clustering, In R. S. Michalski, J, G, Oneli C., and Mite T. M., hell editors, Machine Learning: An Artificial Intelligence Approach", Vol 1, ***Morgan Kaufmann,*** 331-363 (1983).

45. Seidman, C., "Data mining with Microsoft SQL Server 2000", ***Microsoft Press,*** 63 (2000).

46. Agrawal, R., Imielinski, T. and Swami, A., "Mining association rules between sets of items in large databases", ***In ACM SIGMOD Conf. Management of Data,*** 1-10 (1993).

47. İnternet: İTÜ Bilgi İşlem Daire Başkanlığı, "IDS türleri", http://www.bidb.itu.edu.tr/?d=493 (2009).

48. Mukherjee, B., Heberlein, L.T., Levitt, K.N., "Network intrusion detection", ***IEEE Network***, 8(3): 26-41 (1994).

49. Anderson, J.P., "Computer security threat monitoring and surveillance", ***Technical Report, Fort Washington, Pennsylvania,*** 1-30 (1980).

50. Pei, J., Upadhyaya, S.J., Farooq, F., Govindaraju, V., "Data mining for intrusion detection: techniques, applications and systems," ***20th International Conference on Data Engineering (ICDE'04)***, 1063-6382 (2004).

51. Lunt, T.F. "Automated audit trail analysis and intrusion detection: A survey", ***11th National Computer Security Conference,*** Baltimore, MD, 65-73 (1988).

52. Denning, D.E., "An intrusion detection model", ***IEEE Transactions on Software Engineering***, 13(2): 118–131 (1987).

53. Crosbie, M., Spafford, E.H., "Defending a computer system using autonomous agents", ***Technical Report 95-022, Dept. of Comp. Sciences, Purdue University, West Lafayette,*** 1-11 (1995).

54. Endler, D., "Intrusion detection applying machine learning to solaris audit data", ***1998 Annual Computer Security Application Conference (ACSAC'98)***, 268-269 (1998).

55. Axelsson, S., "Intrusion detection systems: A survey and taxonomy", ***Technical Report 99-15, Dept. of Computer Eng., Chalmers University of Technology, Göteborg, Sweden,*** 1-23 (2000).

56. Patcha, A., Park, J.M., "An overview of anomaly detection techniques: Existing solutions and latest technological trends", ***Computer Networks,*** 51(12): 3448-3470 (2007).

57. Endorf, C., Schultz E., Mellander J., "Intrusion detection & prevention", Jenn Tust, Jody McKenzie, Elizabeth Seymour, ***McGraw-Hill***, California, 10-150 (2004).

58. Hofmeyr, S.A., "An immunological model of distributed detection and its application to computer security", Doktora Tezi, ***Computer Science, University of New Mexico***, 1-69 (1999).

59. Dickerson, J.E., Dickerson J.A., "Fuzzy network profiling for intrusion detection" ***NAFIPS 19th International Conference of the North American Fuzzy Information Processing Society***, Atlanta, 301-306, (2000).

60. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "Off-line intrusion detection evaluation data" http://www.ll.mit.edu/IST/ideval/ (2009).

61. İnternet: Knowledge Discovery and Delivery, "KDD Cup 1999: General information",http://www.sigkdd.org/kddcup/index.php?section=1999&method=info (2009).

62. İnternet: MITRE-CWE, Common Weakness Enumeration, "Vulnerability type distributions in CVE", http://cwe.mitre.org/documents/vuln-trends/index.html (2009).

63. Cabrera, B.D., Cabrera, L. Lewis and R.K. Mehra,"Detection and classification of intrusions and faults using sequence of system calls", ***ACM SIGMOD record***, 30(4): 25-34 (2001).

64. Bace, R., Mell, P., "Intrusion detection systems", ***Technical Report, National Institute of Standards and Technology, NIST SP300-31, Scotts Valley, CA,*** 5-46 (2001).

65. Murali, A., Rao, M., "A survey on intrusion detection approaches", ***First International Conference on Information and Communication Technologies***, IEEE Communications Society Press, 233-240 (2005).

66. Mukkamala, S., Janoski, G., Sung, A., "Intrusion detection using neural networks and support vector machines", ***IEEE International Joint Conference on Neural Networks***, IEEE Computer Society Press, 1702-1707 (2002).

67. Şahin, Y. L., "İnternet'te güvenlik ve saldırı sezme sistemleri", Yüksek Lisans Tezi, ***Anadolu Üniversitesi Fen Bilimleri Enstitüsü,*** 1-85 (2005)

68. Fındık, O., Saday, T., "Bilgi güvenliğinin sağlanmasında kullanılan yöntemler ve bunların etkin kullanımı", ***Akademik Bilişim Konferansları***, 1-6 (2003)

69. İnternet: Saldırı tespit sistemleri "Saldırı tespit sistemleri", http://www.tr-security.com/saldiri-tespit-sistemleri/ (2009).

70. Mell, P., Hu, V., Lipmann, R., Haines, J., Zissman, M., "An overview of issues in testing intrusion detection systems", ***Technical Report NIST IR 7007, National Institute of Standard and Technology,*** 1-18 (2003).

71. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "1998 DARPA Intrusion Detection Evaluation Data Set Overview", http://www.ll.mit.edu/IST/ideval/data/1998/1998_data_index.html (2009).

72. İnternet: Massachusetts Teknoloji Enstitüsü Lincoln Laboratuarları "1999 DARPAIntrusion Detection Evaluation Data Set Overview" http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html (2009).

73. Yıldırım, S., "Tümevarım öğrenme tekniklerinden C4.5'in incelenmesi", Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, 1-77 (2003)

74. Özkan, Y., "Veri madenciliği yöntemleri", ***Papatya Yayıncılık Eğitim,*** 1-88 (2008)

75. Silahtaroğlu, G., "Kavram ve algoritmalarıyla temel veri madenciliği", ***Papatya Yayıncılık Eğitim,*** 1-82 (2008)

76. İnternet: C4.5 Algorithm , "C4.5 algorithm", http://en.wikipedia.org/wiki/C4.5 algorithm (2009)

77. Wu, S., Yen, E., "Data mining-based intrusion detectors", ***Elsevier,*** 1-8 (2009)

EKLER

EK-1 Geliştirilen uygulamada kullanılan önemli fonksiyon ve değişkenlerin açıklamaları

int numAttributes: Çıkış niteliği dahil niteliklerin toplam sayısını tutar. Bizim programımızda bu değişkenin değeri 42'dir. Son nitelik karar niteliğini yani sınıflandırmanın sonucunu belirler hangi sınıfa dâhil olduğunu anlamamızı sağlar.

String []attributeNames: Bütün niteliklerin isimlerini tutan dizidir. Son nitelik çıkış niteliğidir.

Vector []domains: Her nitelik için olası değerler bir vektörde saklanmaktadır. domains vektörü numAttributes boyutunda içinde niteliklerin alabileceği değerleri tutan bir dizi vektördür yani bu dizinin her elemanı vektördür ve bu dizi elemanlarının içinde niteliğe karşılık gelen değerleri tutar. Mesela domains[0] vektörü 0. Niteliğe(ilk nitelik) ait değerleri tutar. Aynı şekilde domains vektörünün son elemanı da output niteliğinin sahip olabileceği değerleri tutar.

Class dataPoint: Bu sınıf bir data point temsil eder.int []attributes bu sınıfın bir diğer elemanıdır. Integer tipinden değerler alır. numAttributes boyutunda bir dizidir. Bu dizide niteliklerin bütün değerleri mevcuttur. Dizi sadece sayısal değerler almaktadır. Bu dizinin i. elementi niteliklerin sembolik değerleri bulunan domains vektörüne indeks oluşturur. Mesela attributes[2] = 1 ise 2. niteliğin gerçek / sembolik değeri domains[2].elementAt(1) şeklinde elde edilir. Bu gösterim niteliklerin değerlerinin kıyaslanmasını

EK-1 (Devam) Geliştirilen programda kullanılan önemli değişkenler ve fonksiyonların açıklamaları

kolaylaştırır. Sadece sayısal karşılaştırma yapılır string olarak karşılaştırma yapılamaz. Son nitelik output niteliğidir.

class TreeNode: Ayrıştırma ağacında bir düğüm gösterilirken kullanılacak veri yapısı
double entropy: Yaprak düğümdeki data pointlerin entropisi
public Vector data: Yaprak düğümdeki data point kümesi
public int decompositionAttribute: Hangi niteliğe göre dallandığını gösteren nitelik
public int decompositionValue: Düğümün hangi değere göre dallandığını gösteren değer
public TreeNode []children: Düğümün çocuklarına işaret eder
public TreeNode parent: Düğümün ebeveyni. Kök düğümün ebeveyni = null
TreeNode root = new TreeNode(): Ayrıştırma ağacının kökünün tanımlanması

public int getSymbolValue(int attribute, String symbol): Bu fonksiyon domains vektöründe niteliğin sembolik değerinin bulunduğu indeks değerini integer olarak geri döndürür. Eğer sembol domains vektöründe mevcut değilse bahsi geçen nitelik değerler kümesinin sonuna domains[attribute].addElement(symbol); şeklinde eklenir.

EK-1 (Devam) Geliştirilen programda kullanılan önemli değişkenler ve fonksiyonların açıklamaları

public int []getAllValues(Vector data, int attribute): Belirtilen niteliğin bütün değerlerini int tipinden bir array olarak elde eder.

public Vector getSubset(Vector data, int attribute, int value): Belirtilen nitelik ve belirtilen niteliğin değerine göre veri kümesi döner.

public double calculateEntropy(Vector data): Data Point kümelerinin entropisi hesaplanmaktadır. Entropi, output/çıkış/yaprak niteliklerin değerleri kullanılarak hesaplanır.

public boolean alreadyUsedToDecompose(TreeNode node, int attribute): Belirtilen niteliğin ayrıştırılmasında kullanılıp kullanılmadığını kontrol eder. Bütün ebeveynleri rekürsif olarak kontrolü yapılmaktadır.

public void decomposeNodestandard(TreeNode node): Bu fonksiyon belirlenen düğümü ID3 algoritmasına(standart) göre ayrıştırır.

public void decomposeNodeYeniYontem(TreeNode node): Bu fonksiyon belirlenen düğümü ID3 algoritmasına(önerilen yöntem) göre ayrıştırır.

public void kararAgaciOlusturStandard(): Karar ağacını ID3 algoritması(standart) çerçevesinde oluşturur. Ayrıca fonksiyon içinde çeşitli mesajlar yayarak karar ağacının

EK-1 (Devam) Geliştirilen programda kullanılan önemli değişkenler ve fonksiyonların açıklamaları

oluşturulması aşamasını kullanıcıya bildirir ve Graphviz aracı için gerekli inputlar ve çağrılar yapılarak oluşan karar ağacının resme aktarılmasını sağlar.

public void kararAgaciOlusturYeniYontem(): Karar ağacını ID3 algoritması(önerilen yöntem) çerçevesinde oluşturur. Ayrıca fonksiyon içinde çeşitli mesajlar yayarak karar ağacının oluşturulması aşamasını kullanıcıya bildirir ve Graphviz aracı için gerekli inputlar ve çağrılar yapılarak oluşan karar ağacının resme aktarılmasını sağlar.

public void readFileIntoTextArea(): Veri dosyasını okuyup dosyanın içeriğini TextArea'ya aktaran fonksiyondur. Veri dosyasının ilk satırında niteliklerinin isimleri bulunmalıdır. Niteliklerin sayısı ilk satırdaki kelimelerin sayısı belirlenerek bulunur. Son nitelik ismi output (karar) niteliğidir. Bundan sonraki satırlarda data point için her niteliğe ait değerler bulunmaktadır. Veri dosyası içerisinde satırlar "//" ile başlarsa bu ihmal edilir ve yorum satırı olarak değerlendirilir. Aynı zamanda boş satırlar da dosya okunurken ihmal edilebilir.

testStandard: Bu fonksiyon ile exploreDecisionTree fonksiyonundan gelen değerin saldırı verisinden alınan karar niteliklerinin karşılaştırılması yapılarak eşitlik varsa eşitSayisi adındaki değişken arttırılarak başarım oranı hesaplanır. Bu fonksiyon standart ID3 temel alınarak oluşturulmuş olan ağaçla kıyaslanır.

EK-1 (Devam) Geliştirilen programda kullanılan önemli değişkenler ve fonksiyonların açıklamaları

testYeniYontem: Bu fonksiyon ile exploreDecisionTree fonksiyonundan gelen değerin saldırı verisinden alınan karar niteliklerinin karşılaştırılması yapılarak eşitlik varsa eşitSayisi adındaki değişken arttırılarak başarım oranı hesaplanır. Bu fonksiyon (yeni yöntem) temel alınarak oluşturulmuş olan ağaçla kıyaslanır.

public void exploreDecisionTree: Bu fonksiyonla saldırı verisindeki karar niteliği ile saldırı verisinin karar ağacına giriş olarak verilir ve karar ağacından gelen değerle karşılaştırma yapılır. Buradaki amaç başarım oranının hesaplanmasıdır.

EK-2 Graphviz ile resme aktarılan karar ağacı

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı : USLU, Nurullah Celal
Uyruğu : T.C.
Doğum Tarihi ve Yeri: 16.01.1979, Muğla
Medeni Hali : Bekâr
E-mail : ncuslu@spk.gov.tr

Eğitim

Derece	**Eğitim Birimi**	**Mezuniyet tarihi**
Yüksek lisans	Gazi Üniversitesi	
	Bilgisayar Müh. Bölümü	2009
Lisans	İstanbul Teknik Üniversitesi	
	Bilgisayar Müh. Bölümü	2005
Lise	Ankara Cumhuriyet Lisesi	1999

İş Deneyimi

Yıl	**Yer**	**Görev**
2005-2006	TBMM	Bilgisayar Programcısı
2006-2007	Fintek A.Ş.	Yazılım Uzman Yrd.
2007-2009	T.C. Ziraat Bankası A.Ş.	Müfettiş Yrd.
2009-	Sermaye Piyasası Kurulu	Sistem Mühendisi

Yabancı Dil

İngilizce