
GAZİ ÜNİVERSİTESİ
Gazi
1926

VERİ MADENCİLİĞİ TEMELLİ SİBER TEHDİT İSTİHBARATI

SÜLEYMAN MUHAMMED ARIKAN

YÜKSEK LİSANS TEZİ
ADLİ BİLİŞİM ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM FAKÜLTESİ

NİSAN 2019

Süleyman Muhammed ARIKAN tarafından hazırlanan "VERİ MADENCİLİĞİ TEMELLİ SİBER TEHDİT İSTİHBARATI" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Adli Bilişim Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Dr. Öğr. Üyesi Sami ACAR

Adli Bilişim Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ...……

Başkan: Prof. Dr. Şeref SAĞIROĞLU

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ...……

Üye: Dr. Öğr. Üyesi Bilgin AVENOĞLU

Bilgisayar Mühendisliği Anabilim Dalı, TED Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum ...……

Tez Savunma Tarihi: ...../……/……

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……

Doç. Dr. Aslıhan TÜFEKCİ

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Süleyman Muhammed ARIKAN

08/04/2019

VERİ MADENCİLİĞİ TEMELLİ SİBER TEHDİT İSTİHBARATI

(Yüksek Lisans Tezi)

Süleyman Muhammed ARIKAN

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

NİSAN 2019

ÖZET

Günümüzde siber ortam, her türlü bilgi ve hizmete kolaylıkla erişim için altyapı sağlayarak tartışmasız şekilde insan yaşamının ayrılmaz bir parçası haline gelmiştir. Böylesi önemli bir platformun güvenliği politik, askeri ve stratejik açıdan önem arz etmektedir. Gerek kamu kurumları gerekse özel şirketler siber güvenliği sağlamak için adımlar atmaktadır. Atılan adımların başarıya ulaşmasında siber tehdit istihbaratı büyük öneme sahiptir. Analizler sonucunda açığa çıkarılan siber tehdit istihbaratı saldırgan ve amacı hakkında bilgiler içerir. Ancak, siber tehdit istihbaratı oluşturmak oldukça zahmetli bir iştir. Bu çalışmada veri madenciliği ile siber tehdit istihbaratının üretilmesini kolaylaştıracak bir sistem önerilerek tasarımı yapılmış ve uygulması geliştirilmiştir. Önerilen sistem ile kaydedilmiş trafik kayıtları veya canlı trafik kayıtları öğrenilmiş saldırı tiplerine göre sınıflandırılarak siber tehdit istihbaratı yaygın bir standart olan STIX formatında üretilmektedir. Sistem, bilinmeyen saldırıların uzman görüşü ile tanımlanmasına olanak sağlayarak öğrenme kümesini yeni saldırı tipleri ile güncelleyebilmektedir. Çalışmada öncelikle, saldırı tiplerinin öğrenilebilmesi için kullanılması muhtemel çeşitli veri kümeleri incelenmiştir. Kullanılan veri madenciliği algoritmalarının başarımını karşılaştırabilmek amacıyla genel kabul görmüş KDD CUP 99 ve %10 KDD CUP 99 veri kümeleri tercih edilmiştir. İlgili veri kümelerinin birçok dezavantajı bulunduğundan dolayı NSL-KDD veri kümesi, siber tehdit istihbaratı üretmek için ihtiyaç duyulan saldırı tiplerini öğrenmek için kullanılmıştır. Trafik kayıtlarını sınıflandırmak için veri madenciliği tekniği olarak karar ağacı algoritmalarından CART ve ID3 kullanılmıştır. %10 KDD CUP 99 ve NSL-KDD veri kümeleri üzerinde ilgili algoritmalar için tabakalı 10 katlamalı çapraz doğrulama yapılarak literatürde yer alan çalışmalar ile kıyaslanmıştır. Tabakalı 10 katlamalı çapraz doğrulama için %10 KDD CUP 99 veri kümesi üzerinde sırası ile CART ve ID3 algoritmaları için başarımlar %99,9 ile %99,4 olarak gözlemlenirken NSL-KDD veri kümesi üzerinde ise sırası ile CART ve ID3 algoritmaları için başarımlar %98,8 ile %96,2 olarak ölçülmüştür. Buna ek olarak benzer adımların veri madenciliğinde güvenilirliğini kanıtlamış Weka aracı üzerinde gerçekleştirilmesi ile elde edilen başarımların önerilen sistemin başarımlarına olan benzerliği gösterilmiştir. Önerilen sistemin uygulaması için örnek bir çalışma olarak 24 saat boyunca internete açık çalıştırılan bir sunucu üzerinde bal küpü sistemleri ile saldırganların dikkati çekilerek ilgili trafik kayıtları kaydedilmiş ve kayıtlar üzerinden siber tehdit istihbaratı oluşturulmuştur. Önerilen sistemin siber tehdit istihbaratı üretmek için başarılı bir şekilde kolaylıkla kullanılabileceği gösterilmiş ve internet ortamına açılacak olan sunucular için kullanılabilecek siber tehdit istihbaratı üretilmiştir.

Bilim Kodu : 92403
Anahtar Kelimeler : Veri madenciliği, saldırı tespit sistemi, siber tehdit istihbaratı
Sayfa Adedi : 127
Danışman : Dr. Öğr. Üyesi Sami ACAR

DATA MINING BASED CYBER THREAT INTELLIGENCE

(M. Sc. Thesis)

Süleyman Muhammed ARIKAN

GAZİ UNIVERSITY

INFORMATICS INSTITUTE

APRIL 2019

ABSTRACT

Nowadays, cyber environment has become an integral part of human life by providing the infrastructure for easy access to all kinds of information and services. Security of such an important platform is of political, military and strategic importance. Both public institutions and private companies are taking steps to ensure cyber security. Cyber threat intelligence has a great importance for success of the steps taken. The cyber threat intelligence revealed by the analyzes contains information about attacker and his/her purpose. However, creating cyber threat intelligence is very demanding task. In this study, a system that will facilitate the creating cyber-threat intelligence with data mining is proposed, designed and its application developed. With the proposed system, saved or live traffic records can be classified according to learned attack types and the cyber threat intelligence can be produced in STIX format which is a common standard. The system is able to update training set with new attack types by allowing unknown attacks to be identified by expert opinion. First of all, various datasets which are likely to be used to learn the types of attack are examined. In order to compare performance of data mining algorithms, KDD CUP 99 and 10% KDD CUP 99, which are generally accepted datasets, were preferred. Since these datasets have many drawbacks, NSL-KDD dataset is used to learn the types of attacks needed to generate cyber threat intelligence. In order to classify traffic records, CART and ID3 from decision tree algorithms were used as data mining techniques. 10% KDD CUP 99 and NSL-KDD were compared to studies in literature by using stratified 10-fold cross validation for the algorithms on the datasets. For stratified 10-fold cross-validation, accuracy of CART and ID3 algorithms on the 10% KDD CUP 99 dataset was 99.91% and 99.4%, respectively. On the NSL-KDD dataset, accuracy of CART and ID3 algorithms was 98,76% and 96.18%, respectively. In addition, similarity of accuracy by performing similar steps in Weka tool, which has proven its reliability in data mining, to the accuracy of the proposed system was demonstrated. As a sample work for application of proposed system, a server was run for 24 hour with honeypot system which is for drawing attention of attackers. Then, server traffic were recorded and this records is used for generating cyber threat intelligence. It has been shown that the proposed system can be easily used successfully to generate cyber threat intelligence and cyber threat intelligence has been produced for servers that will be opened to the internet.

Science Code : 92403
Key Words : Data mining, intrusion detection system, cyber threat intelligence
Page Number : 127
Supervisor : Assist. Prof. Dr. Sami ACAR

TEŞEKKÜR

Değerli yardım ve katkılarıyla çalışmalarımı yönlendiren tez danışmanım Sayın Dr. Öğr. Üyesi Sami ACAR'a, jüri üyeleri Sayın Prof. Dr. Şeref SAĞIROĞLU'na ve Sayın Dr. Öğr. Üyesi Bilgin AVENOĞLU'na, çalışmamın başarıya ulaşması için her türlü bilgisini benimle paylaşmaktan çekinmeyen tüm TÜBİTAK BİLGEM Siber Güvenlik Enstitüsü mesai arkadaşlarıma ve siber tehdit istihbaratı ile veri madenciliği konuları üzerine bilgi ve birikim kazanmamda büyük öneme sahip olan Sayın Başuzman Araştırmacı Özgür YÜREKTEN'e ve Sayın Araştırmacı Tolga YILMAZ'a, bana olan inançlarını hiç kaybetmeyen ve destekleriyle beni hiçbir zaman yalnız bırakmayan sevgili aile üyelerim annem Nurşen ARIKAN'a, babam Ergin ARIKAN'a, kardeşlerim Muhammed Yasin ARIKAN'a ve Nehir ARIKAN'a, kayınvalidem Gülnaz DOLAŞIK'a, kayınpederim Ahmet DOLAŞIK'a, ablam Çiğdem DOLAŞIK'a, çalışma süresince tüm zorlukları benimle göğüsleyen ve her daim bana destek olan değerli eşim Cansu ARIKAN'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

İÇİNDEKİLER vii

ÇİZELGELERİN LİSTESİ x

ŞEKİLLERİN LİSTESİ xi

RESİMLERİN LİSTESİ xii

KISALTMALAR xiv

1. GİRİŞ 1

2. TEMEL BİLGİLER, KAVRAMLAR VE TERİMLER 11

2.1. Veri Madenciliği 11

2.1.1. Veri tabanından bilgi keşfi süreçleri 11

2.1.2. Veri madenciliğinde dikkat edilmesi gereken hususlar 14

2.1.3. Veri madenciliğinde kullanılan modeller ve algoritmalar 16

2.1.4. Veri tabanından bilgi keşfi uygulama alanları 28

2.2. Saldırı Tespit Sistemi 29

2.2.1. Saldırı tespit sistemi sınıflandırmaları 30

2.2.2. Saldırı tespit sisteminde kullanılan yöntemler 32

2.2.3. Saldırı tespit sistemleri sınırlılıkları 33

2.3. Bal Küpü Sistemleri 33

2.3.1. Bal küpü sistemi sınıflandırmaları 34

2.3.2. Bal küpü sistemlerinin önemi 34

2.4. Siber Tehdit İstihbaratı 35

Sayfa

2.4.1. Siber tehdit istihbaratı sınıflandırmaları 35
2.4.2. Siber tehdit istihbaratı standartları 36
2.4.3. Siber tehdit istihbaratının önemi 40
2.4.4. Siber tehdit istihbaratı paylaşmak için platformlar 41
2.4.5. Siber tehdit istihbaratı elde etmede yaşanan zorluklar 42
3. ÖNERİLEN SİSTEM TASARIMI 43
3.1. Genel Sistem Tasarımı 43
3.1.1. Kayıt modülü 44
3.1.2. Raporlama modülü 44
3.1.3. Saldırı tespit sistemi modülü 45
3.1.4. Ağ trafiği dinleme modülü 46
3.1.5. Zafiyet simülasyon modülü 46
3.1.6. Tehdit simülasyon modülü 47
3.2. Saldırı Veri Kümeleri 47
3.2.1. Mevcut veri kümeleri üzerinde gerçekleştirilen incelemeler 47
3.2.2. Seçilen veri kümeleri 49
4. ÖNERİLEN SİSTEMİN UYGULAMASI 55
4.1. Geliştirilen Modüller 58
4.1.1. Kayıt modülü 58
4.1.2. Raporlama modülü 61
4.1.3. Saldırı tespit sistemi modülü 68
4.1.4. Ağ trafiği dinleme modülü 70
4.1.5. Zafiyet simülasyon modülü 71
4.1.6. Tehdit simülasyon modülü 72
4.2. Uygulanan Veri Tabanından Bilgi Keşfi Süreçleri 72

Sayfa

4.2.1. Veri seçimi adımı....72
4.2.2. Önişleme adımı....76
4.2.3. Dönüştürme adımı....78
4.2.4. Veri madenciliği adımı....82
4.2.5. Bilginin sunumu adımı....97
4.3. Önerilen Sistem İle Siber Tehdit İstihbaratı Üretme....98
4.3.1. Veri toplama....98
4.3.2. Siber tehdit istihbaratının üretilmesi....100
5. SONUÇ VE ÖNERİLER....103
KAYNAKLAR....108
ÖZGEÇMİŞ....127

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. STIX bileşenleri ……..... 39

Çizelge 2.2. STİ üretmede yaşanan zorluklar …… 42

Çizelge 3.1. KDD CUP 99 saldırı kategorileri ve barındırdıkları saldırılar …..... 50

Çizelge 3.2. NSL-KDD Eğitim - protokol düzeyinde saldırı kayıtları dağılımı …..... 53

Çizelge 4.1. Modüller ve sistem kaynakları ……..... 58

Çizelge 4.2. Raporlama modülü kullanılan kütüphaneler 61

Çizelge 4.3. Saldırı tespit sistemi modülü kullanılan kütüphaneler 69

Çizelge 4.4. Kategorik değerlerin sayısal değerler ile değiştirilmesi ……...…….. 77

Çizelge 4.5. %10'luk örneklem üzerinde sonuçların karşılaştırılması 83

Çizelge 4.6. %10'luk örneklem üzerinde başarım kıyaslaması 83

Çizelge 4.7. %2'lik örneklem üzerinden sonuçların karşılaştırılması 88

Çizelge 4.8. %2'lik örneklem üzerinden başarım kıyaslaması 89

Çizelge 4.9. Farklı port sayısı ve saldırı hızı ile portsweep başarımları 93

Çizelge 4.10. NSL-KDD veri kümesi algoritmaların kıyaslanması 96

Çizelge 4.11. İnternete açık sunucu sistem kaynakları ……..... 99

Çizelge 4.12. Yakalanmış trafik kayıtlarının sınıflandırma sonuçları …..... 100

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 2.1. Veri tabanından bilgi keşfi süreçleri 11

Şekil 2.2. Karar ağacında bulunan karar düğümleri ve sınıf yaprakları 17

Şekil 2.3. Yapay sinir ağları katmanları 21

Şekil 2.4. Destek vektör makineleri için hiperdüzlemler ve maksimum tolerans 22

Şekil 2.5. Hiyerarşik kümeleme için örnek dendogram 25

Şekil 2.6. TAXII protokolü istemci – sunucu mimarisi 37

Şekil 3.1. Önerilen sistem için bileşen diyagramı 44

Şekil 4.1. Uygulama dosya ve dizin yapısı …..... 62

RESİMLERİN LİSTESİ

Resim **Sayfa**

Resim 2.1. STIX 1.0 versiyonu ile üretilmiş STİ 37

Resim 2.2. CyBOX ile üretilmiş STİ 38

Resim 2.3. STIX 2.0 versiyonu ile üretilmiş STİ 38

Resim 2.4. Tool, Vulnerable ve Indicator STIX bileşenleri için örnek kayıtları 39

Resim 2.5. VERIS ile üretilmiş STİ 40

Resim 4.1. Gerçek trafik kayıtlarından STİ elde edilmesi için akış diyagramı 57

Resim 4.2. Raporlama modülü anasayfası 64

Resim 4.3. Saldırı tespit sistemi sayfası 65

Resim 4.4. Öğrenenilmiş ağaca raporlam uygulaması üzerinden ulaşma 65

Resim 4.5. Bilinmeyen saldırı kayıtları sayfası 66

Resim 4.6. Gerçek trafik kayıtları sayfası 67

Resim 4.7. Üretilmiş STİ için sunulan formatlar 67

Resim 4.8. Farklı formatlarda üretilmiş STİ 68

Resim 4.9. Siber tehdit istihbaratları sayfası 68

Resim 4.10. Eğitim kümesi oluşturma adımı - kayıtların getirilmesi 73

Resim 4.11. Eğitim kümesi oluşturma adımı - kayıt sayılarının onaylanması 73

Resim 4.12. Eğitim kümesi oluşturma adımı - örneklem çıkarma 74

Resim 4.13. Test kümesi oluşturma adımı 74

Resim 4.14. Kaydedilmiş trafik kayıtlarının veri tabanına kaydedilmesi 75

Resim 4.15. Canlı trafik üzerinden yakalanan kayıtlar 76

Resim 4.16. Sapan veri analizi adımı 78

Resim **Sayfa**

Resim 4.17. Veri formatı dönüştürme adımı ……..……..... 78

Resim 4.18. Öznitelikler ve sınıf etiketi arasındaki ilişki kuvvetleri grafiği ……... 80

Resim 4.19. Öznitelikler ve sınıf etiketi arasındaki ilişki kuvvetleri dereceleri …….. 81

Resim 4.20. Sürekli verilerin gruplanması 81

Resim 4.21. KDD CUP 99 veri kümesi üzerinde veri madenciliği akış diyagramı 82

Resim 4.22. Tabakalı 10 katlamalı çapraz doğrulama için akış diyagramı ……..... 85

Resim 4.23. Algoritmaların veri kümelerine göre başarımları …..... 86

Resim 4.24. Öğrenme adımı için kullanılacak algoritmanın belirlenmesi ……..…… 88

Resim 4.25. Test kümesi sınıflandırma adımı ……..... 90

Resim 4.26. Özniteliklerin seçim kutuları ile belirlenme adımı ……..…..... 91

Resim 4.27. KDD CUP 99 %10 veri kümesi çapraz doğrulama sonuçları ……..…….. 91

Resim 4.28. NSL-KDD eğitim kümesi saldırı tipleri güncellik testi adımı 92

Resim 4.29. NSL-KDD Eğitim veri kümesi çapraz doğrulama sonuçları …..... 94

Resim 4.30. NSL-KDD veri kümesi ID3 algoritması başarımları ……..... 95

Resim 4.31. NSL-KDD veri kümesi CART algoritması başarımları ……..... 95

Resim 4.32. Gerçek trafik kayıtlarını sınıflandırma adımı ……..... 97

Resim 4.33. CART algoritması – ilgili karar ağacının örüntüsü ve kesiti …….. 97

Resim 4.34. Sektörlere göre günlük ortalama web saldırıları ……..... 98

Resim 4.35. İlk 24 saatlik periyot için gelen trafik paketleri ortalama sayısı ……. 99

KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
STS	Saldırı Tespit Sistemi
STİ	Siber Tehdit İstihbaratı
KSU	Komut Satırı Uygulaması
DOS	Denial Of Service
STIX	Structured Threat Intelligence Expression
CYBOX	Cyber Observable Expression
VERIS	Vocabulary for Event Recording and Incident Sharing
TAXII	Trusted Automated Exchange of Intelligence Information
MISP	Malware Information Sharing Platfrom
IBM	International Business Machines Corporation
YSA	Yapay Sinir Ağları
GP	Gerçek Pozitif
YP	Yanlış Pozitif
GN	Gerçek Negatif
YN	Yanlış Negatif

1. GİRİŞ

Her türlü bilgi ve hizmete kolaylıkla erişim için altyapı sağlayan ve tartışmasız şekilde insan yaşamının ayrılmaz bir parçası haline gelen siber ortam, güvenliği bakımından politik, askeri ve stratejik önem arz etmektedir. Kamu kurumları, özel şirketler, kar amacı gütmeyen organizasyonlar ve bireyler siber ortamın güvenliğini sağlamak için adımlar atmaktadır. Siber tehdit istihbaratı (STİ), bu adımların başarıya ulaşması için büyük öneme sahiptir.

STİ, siber tehditlerin tanımlanabilmesi ve yakalanabilmesi için gerekli olan bilgi veya siber güvenliğin sağlanabilmesi için kullanılabilecek her türlü veri olarak tanımlanabilir [1, 2]. STİ ile mevcut saldırılar hakkında bilgi sahibi olunabileceği gibi yeni veya yaklaşmakta olan saldırılar hakkında da veri elde edilebilir. Bu sayede maruz kalınan saldırılar için önlemler alınabilmekte ya da maruz kalınabilecek saldırılar için engelleyici faaliyetler geliştirilebilmektedir [3]. Bu sayede birçok saldırıdan henüz gerçeklemeden dahi korunulabilir.

STİ, Birleşik Krallık Ulusal Siber Güvenlik Merkezi tarafından 4 gruba ayrılmıştır [4]. Barındırılan bilgiye ve ilgili bilgilinin hedef kitlesine göre yapılan bu gruplama ile taktiksel, teknik, operasyonel ve stratejik siber tehdit istihbaratları tanımlanmıştır [5]. Saldırganların kullandığı araçlar ve yöntemleri hakkında bilgiler taktiksel STİ ile, saldırgan veya zararlı yazılımı tespit etmek için gerekli bilgiler teknik STİ ile, finansal konularda veya saldırı eğilimleri hakkında bilgiler stratejik STİ ile ve organizasyonlara gerçekleştirilebilecek saldırılar hakkında bilgiler ise operasyonel STİ ile paylaşılır [5].

STİ için oluşturulmuş birçok standart mevcuttur. Bu standartlar; STİ'nin kolay bir şekilde anlaşılması, oluşturulması, paylaşılması ve kullanılması için geliştirilmiştir. Böylece tüm paydaşlar arasında ortak bir dil oluşturulur. İlgili standartlara Structured Threat Intelligence Expression (STIX), Cyber Observable Expression (CyBOX), Vocabulary for Event Recording and Incident Sharing (VERIS) ve Trusted Automated Exchange of Intelligence Information (TAXII) örnek olarak gösterilebilir [6].

İlgili standartlar kullanılarak üretilmiş STİ'nin, diğer kullanıcılar tarafından kolayca erişilmesini ve faydalanılmasını sağlamak için birçok platfrom kullanılmaktadır. Bu platformlara Malware Information Sharing Platform (MISP) ve International Business Machines Corporation X-Force Exchange (IBM X-Force Exchange) örnek olarak

gösterilebilir [7]. İlgili platfromlar kullanılarak oluşturulmuş bir STİ, kolaylıkla paylaşılabilir ve paydaşlara ulaştırılabilir.

Literatür taramasında Eric Nunes ve arkadaşları tarafından gerçekleştirilen bir çalışmada darknet ve deepnet gibi kavramlarla ifade edilen ulaşılabilir olmayan gizli internet kaynakları gibi sosyal platformlar üzerinde bulunan forumlardan ve marketlerden elde edilen bilgiler kullanılarak çeşitli veri madenciliği teknikleri ile STİ toplandığı görülmüştür [8]. Çalışma kapsamında oluşturulmuş ilgili sistem; toplayıcı (crawler), ayrıştırıcı (parser) ve sınıflandırıcı (classifier) olmak üzere 3 adet bileşenden oluşmaktadır. Toplayıcı ile forumlar ve marketler üzerinden web sayfası kayıtları elde edilir. Ayrıştırıcı bileşeni ile bu kayıtlar üzerinden belirli bilgiler çıkartılır ve veri tabanına kaydedilir. Sınıflandırıcı bileşeni ile veri madenciliği tekniklerinden faydalanılarak ilgili kayıtlar sınıflandırılır. Sınıflandırma işlemi için naive bayes sınıflandırıcısı, random forest, destek vektör makineleri ve lojistik regresyon karşılaştırılmıştır. İlgili karşılaştırmalar sonucunda market verilerini sınıflandırma işlemi için en yüksek doğruluğa destek vektör makineleri ile ulaşılırken forum verilerini sınıflandırma işlemi için ise en yüksek doğruluk lojistik regresyon ile elde edilmiştir. Sınıflandırma işlemlerinin sonucunda STİ elde edilir. Ancak ilgili sistem, elde edilmiş istihbarat bilgilerini herhangi bir STİ standardına uygun olarak üretmemektedir.

Bu çalışmada üretilmiş STİ için kaynak olarak gerçek trafik kayıtları kullanılmıştır. Bu kapsamda saldırı tiplerinin öğrenilmesi ve trafiğin sınıflandırılması için saldırı tespit sisteminden (STS) faydalanılmıştır. Bir STS, zararlı aktivitelerin tespiti için ağ trafiğini sürekli olarak izleyen sistemdir [9]. İzlenen bu trafik üzerinde yapılan analizler ile şüpheli durumlar rapor edilerek alarm üretilmektedir [10]. Dolayısı ile asıl amaç zararlı aktivitelerin yakalanması ve raporlamadır. Bulundukları konumlara göre ağ tabanlı ve anasistem tabanlı olarak STS'ler iki sınıfa ayrılabilir. Ağ tabanlı STS ile ağ üzerinden akan trafik üzerinde analizler geçekleştirilirken anasistem tabanlı STS ile yalnızca kurulu olduğu cihazın trafiği üzerinde analizler yapılmaktadır [11].

STS'ler kural tabanlı veya anomali tabanlı olabilir [12]. Kural tabanlı bir STS'nin zararlı aktiviteyi tespit edebilmesi için ilgili saldırı tipi için önceden bir kuralın oluşturulması gerekmektedir [13]. Bu oldukça zor bir işlemdir. Ayrıca birçok saldırı tipi yalnızca bir kural ile tespit edilememektedir. Buna ek olarak anomali tabanlı STS'ler ise bilinmeyen saldırıların tespit edilmesinde oldukça faydalıdır [14]. Trafiğin olağan durumunu öğrenerek

aykırı durumlar üzerinden yapılan analizler ile zararlı aktivitelerin tespiti gerçekleştirilmektedir. Ancak oldukça fazla yanlış alarm ile karşılaşılabilmektedir [15].

Veri madenciliği teknikleri, hem kural tabanlı STS hemde anomali tabanlı STS performansını geliştirmek için kullanılabilir. Sınıflandırma, kümeleme ve ilişki kuralları gibi veri madenciliği teknikleri saldırılar hakkında bilgi toplamak için sıklıkla tercih edilmektedir[16].

Literatür taramasında Kailas Shivshankar Elekar tarafından ele alınan bir çalışma ile anomali tabanlı STS'lerin yüksek sayıdaki yanlış alarm sayısını düşürmek için birden fazla veri madenciliği tekniğini aynı anda kullandığı görülmüştür [17]. İlgili çalışmada J48, random forest ve random tree algoritmaları üzerinde çalışılmıştır. Sırası ile J48 ve random forest, J48 ve random tree, random forest ve random tree algoritmaları beraber kullanılarak sonuçlar karşılaştırılmıştır. Öncesine nazaran daha az yanlış alarm üretildiğinin görülmesine ek olarak birlikte çalıştırılan algoritmalar arasında J48 ve random tree algoritmalarının beraber kullanımının diğerlerine göre daha verimli olduğu görülmüştür.

Kathleen Goeschel tarafından hazırlanan bir çalışmada geleneksel STS yaklaşımlarının yüksek sayıda yanlış alarm üretmesinin önüne geçmek ve daha farklı bir yaklaşım oluşturmak amacı ile birden çok veri madenciliği algoritmasını birlikte kullanarak doğru tespit oranının artırılması sağlanmıştır [18]. İlgili çalışmada destek vektör makinesi, karar ağacı ve naive bayes sınıflandırıcısı kullanılarak bir model oluşturulmuştur. Amaçlanan modelin gerçekleştirilebilmesi adına destek vektör makinesi için LibSVM [19] kullanılmış ve karar ağacı olarak J48 tercih edilmiştir. İlk olarak LibSVM ile ilgili kaydın bir saldırı trafiğine mi yoksa normal bir trafiğe mi ait olduğuna karar verilmesi için çalışma yapılmıştır. Saldırı trafiği olarak belirlenen kayıt J48 algoritmasına yönlendirilir ve saldırının türü belirlenir. Eğer saldırı türü önceden öğrenilmiş değilse ve türü bilinemiyorsa bu defa ilgili kayıt karar ağacından naive bayes sınıflandırıcısına gönderilerek kaydın diğer saldırılarla olan benzerliği hesaplanır ve en olası saldırı türü belirlenir.

Literatür taramasında ayrıca Shankar M. Banik ve arkadaşlarının ağ trafiğini izlemek yerine işletim sistemi günlüklerini analiz ederek bazı saldırı tiplerini yakalama amacında olduğu görülmüştür [20]. İlgili çalışmada yetkisiz veya zararlı aktivitelerin tespiti için Ubuntu işletim sistemi günlük dosyaları incelenmiştir. Veri madenciliği tekniklerinden kümeleme algoritmalarının kullanıldığı bu çalışmada kaba kuvvet saldırılarının ve servis

dışı bırakma saldırılarının yakalanmasında verimli sonuçlar verdiği görülmüştür. Ancak işletim sistemi günlükleri üzerinden saldırıların çok azı hakkında bilgi sahibi olunacağından birçok saldırıyı destekleyecek bir STS için ilgili yöntemin faydalı olmadığı değerlendirilmiştir.

Dikshant Gupta ve arkadaşları doğrusal regresyon algoritması ile K ortalama algoritmasını kullanarak ağ paketlerini sınıflandırmak için kural üretme amacıyla bir çalışma gerçekleştirmiştir [21]. Bu çalışmada NSL-KDD veri kümesi [22] kullanılmıştır. Doğrusal regresyon algoritmasının sürpriz şekilde daha iyi sonuç verdiği görülmüştür. K ortalama algoritmasının %67,5 doğruluk ile çalışmasına karşın doğrusal regresyon %80 doğruluk ile çalıştığı gösterilmiştir.

Nadya El Moussaid ve Ahmed Toumanari geleneksel yaklaşımlar kullanan STS'lerin bilinen ve bilinmeyen saldırıların tespitindeki performanslarının geliştirilmesi için veri madenciliği tekniklerinden faydalanmayı amaçlanmıştır [23]. KDD CUP 99 veri kümesi kullanılarak gerçekleştirilen bu araştırmada K ortalama algoritması, Y ortalama algoritması ve grafik tabanlı kümele algoritması kullanılmıştır. Edinilen sonuca göre en kolay ve basit kümeleme algoritması olan k ortalama algoritmasının gürültüden ve sapan veriden fazla etkilenmesine karşın yüksek tespit oranına ve düşük yanlış alarm sayısına sahip olduğu görülmüştür. Y ortalama algoritmasının da yüksek tespit oranına ve düşük yanlış alarm sayısına sahip olmasına karşın gerçek zamanlı sistemlerde saldırıların tespit edilmesi için kullanılmasında birçok zorlukla karşılaşılmıştır. Tüm bunlara ek olarak aynı çalışmada incelenen bir diğer veri madenciliği algoritması olan grafik tabanlı kümele algoritmasının da yüksek tespit oranına ve düşük yanlış alarm sayısına sahip olmasına karşın oldukça fazla işlemci gücüne ihtiyaç duyduğu görülmüştür.

Şeref Sağıroğlu ve arkadaşları tarafından ele alınan bir çalışmada genel olarak klasik ve zeki STS'ler incelenmiş, STS tasarımında kullanılan veri kümeleri değerlendirilmiş ve yapay sinir ağları (YSA) tabanlı zeki bir STS tasarımı yapılmıştır [24]. Geliştirilen STS ile farklı eğitim ve test kümeleri kullanılarak testler gerçekleştirilmiştir. İlgili testler sonucunda en yüksek başarım oranı %97,92 olarak ölçülürken en düşük başarım oranının ise %81,93 olduğu görülmüştür. İlgili çalışmada önerilen modelin, literatürde başarılı kabul edilmiş sistemlerden daha fazla sayıda saldırı tipini daha kısa sürede tanıyabildiği sonucuna ulaşılmıştır. Ayrıca araştırmacıların, STS uygulamalarında kullanılacak veri kümesini seçerken literatürde kabul görmüş veri kümeleri yerine kendi oluşturacakları veri

kümesini seçmeyi tercih ettiklerinde oldukça zahmetli ve maliyetli bir çalışma gerçekleştirmenin zorluklarının ortaya çıkabileceğinden bahsedilmiştir. Bu kapsamda birçok veri kümesi incelenmiş ve geliştirilen sistemin objektif olarak değerlendirilebilmesi için saldırı veri tabanı olarak KDD CUP 99 veri kümesi seçilmiştir. Tüm bunlara ek olarak ilgili çalışma ile zeki yaklaşımların STS tasarımında kullanılmasının doğru bir yaklaşım olduğu, MLP yapısının LM gibi güçlü algoritmalarla eğitilmesi ile YSA'ların hesaplama süresini düşürebileceği ve farklı saldırı tiplerini belirlemede başarıyı artıracağı gibi sonuçlara ulaşılmıştır.

Subaira ve Anitha veri madenciliği temelli bir STS içinde kullanılabilecek algoritmaları karşılaştırmıştır [25]. Bu çalışmada destek vektör makineleri, sinir ağları, bulanık mantık, bayes sınıflandırıcısı, k-en yakın komşu ve karar ağacı algoritmaları kullanılmıştır. İlgili algoritmalar bir STS ile kullanılarak her birinin sonuçları üzerinden karşılaştırma yapılmıştır. Çalışmaya göre destek vektör makineleri yüksek karmaşıklık içermesine ek olarak aşırı bellek kullanımına ihtiyaç duymaktadır. Ayrıca öğrenme ve test süresinin oldukça uzun olduğu görülmüştür. Genetik algoritmaları ile aynı problemin farklı yollardan birçok çözümünün bulunabilmesine ek olarak öğrenilmiş bilgilerin kolaylıkla transfer edilebileceği görülmüştür. Ancak sabit bir süre içinde yanıt verememiştir. Dolayısı ile farklı zaman aralıklarında sonuçlar dönmüştür. İlgili çalışma, k-en yakın komşu algoritmasının yüksek kayıt alanına ihtiyaç duyduğu, öğrenme ve test sürelerinin uzun olduğu ve veri boyutundan oldukça fazla etkilendiği sonuçlarına ulaşmıştır. Sinir ağları ile bağımlı ve bağımsız öznitelikler arasındaki doğrusal olmayan karmaşık ilişkilerin bulunmasında oldukça etkili olduğu görülmüştür. Ancak aynı çalışmada sinir ağlarının yüksek hesaplama gücüne ihtiyaç duyduğundan öğrenme süresinin oldukça uzun olduğu tespit edilmiştir. Sonuçları incelenen diğer bir algoritma olan bulanık mantığın bulanık eşik değerleri ile işlem yapmaya imkân sağlamasına ek olarak bulanık sistemden bir model üretmenin zor olduğu sonucuna ulaşılmıştır. Tüm bunlara ek olarak, aynı çalışmada karar ağacı algoritmaları sonuçları da incelenmiştir. Karar ağacı oluşturmak için çalışılan alan hakkında bilgi sahibi olunmasına ihtiyaç duyulmaması, yüksek boyutlu veri ile dahi çalışabilmesi, gösteriminin oldukça kolay bir şekilde anlaşılabilir olması ve hem sayısal hem de kategorik veri ile verimli bir şekilde sonuç üretebilmesi karar ağacı algoritmaları hakkında elde edilmiş bilgilerdir.

Bu çalışmada STS tarafından sınıflandırılacak trafik kayıtlarının elde edilmesi için bal küpü sistemlerinden faydalanılmıştır. Bal küpü sistemleri, saldırganların dikkatini çekerek saldırıları kendilerinde yönlendiren ve tuttuğu günlükler ile saldırganlar ve teknikleri hakkında bilgiler toplayan sistemlerdir [26]. Bal küpü sistemleri ile saldırganların dikkati çekilerek zararlı trafik kaydedilebilir ve bir uzman tarafından ilgili kayıtlar üzerinde detaylı analiz yapılabilir. Ayrıca kurulu oldukları cihazı, zafiyetli olarak göstermelerine ek olarak ilgili zafiyeti simüle ederek saldırganların faaliyetleri hakkında detaylı bilgi toplanabilir. Bu sistemler, adını ayı tuzaklarını cazip hale getirmek için kullanılan bal küplerinden alır [26].

Bal küpü sistemleri düşük etkileşimli ve yüksek etkileşimli olmak üzere iki sınıfa ayrılmaktadır [27]. Düşük etkileşimli bal küpü sistemleri, gerçek servisleri barındırmak yerine ilgili servisleri yalnızca simüle eder [26]. Yüksek etkileşimli bal küpü sistemleri ise düşük etkileşimli bal küpü sistemlerine nazaran daha karmaşık yapıya sahiptirler. Gerçek servis ve sistemler barındırılır [26]. Düşük etkileşimli bal küpü sistemlerine Honeyd ve Dionaea örnek olarak gösterilebilirken yüksek etkileşimli bal küpü sistemlerine HoneyBrid ve Sebek örnek olarak verilebilir.

Bu çalışmada problem, STİ üretmenin zahmeti bir iş olması ve bu zahmetten dolayı birçok araştırmacının ve/veya kurumun STİ üretememesi olarak ele alınmıştır. STİ üretmek için ilgili alan hakkında uzman bilgisine ihtiyaç duyulmaktadır. Yeterli uzman bilgisine ulaşılamadığı durumlarda STİ elde edilememekte veya yanlış değerlendirilmiş bilgiler içerebilmektedir. Bunun yanı sıra, uzman bilgisine sahip olunduğu durumlarda STİ üretme aşamalarının hemen hepsi manuel işlemler içerdiğinden oldukça uzun ve zahmetli bir çalışma sürecine ihtiyaç duyulmaktadır. Ayrıca, saldırıların tespiti için kullanılan STS'ler tarafından üretilen raporlar arasında genel bir benzerlik yoktur [28]. Dolayısı ile her STS'nin çıktısı farklı bir veri modeli barındırmaktadır. Birden çok STS ile çalışılan durumlarda her bir STS raporunun veri modeli iyi bilinmeli ve bu farklı veri modellerinden kazanılan bilgiler ile STİ üretilebilmelidir. Bu durum STİ oluşturulmasını zorlaştırmaktadır. STS'ler tarafından otomatik olarak üretilmediğinden dolayı kullanıcılar tarafından oluşturulmak zorunda olan STİ, oldukça fazla iş yükü gerektirmekte ve birçok şirket ve/veya kurum tarafından bu iş için ayrılan zaman gereksiz görülmektedir. Gerçekleştirilen bu çalışma kapsamında STİ oluşturmada kullanılan manuel adımların otomatize edilmesi ve uzman bilgisine ihtiyaç duyulmaksızın kolaylıkla STİ üretebilme

yeteneğinin kullanıcılara kazandırılması amaçlanmıştır. Bu sebeple, veri madenciliği ile STİ üretilmesini kolaylaştıracak bir sistem önerilmiştir.

Önerilen sistem ile kaydedilmiş trafik kayıtları veya canlı trafik kayıtları öğrenilmiş saldırı tiplerine göre sınıflandırılarak siber tehdit istihbaratı yaygın bir standart olan STIX formatında üretilmektedir. Sistem, bilinmeyen saldırıların uzman görüşü ile tanımlanmasına olanak sağlayarak öğrenme kümesini yeni saldırı tipleri ile güncelleyebilme yeteneğine sahiptir.

Çalışmada öncelikle, bir STS'de bulunması gereken bileşenler araştırılmıştır. Elde edilen bilgiler ışığında önerilen sistemin modüllerinin tasarımı yapılmıştır. Önerilen sistem; kayıt modülü, raporlama modülü, saldırı tespit sistemi modülü, ağ trafiği dinleme modülü, zafiyet simülasyon modülü ve tehdit simülasyon modülü olmak üzere 6 adet modülden oluşmaktadır. Kayıt modülü ile STS için saldırı veri kümeleri, veri madenciliğinde kullanılan parametreler, STİ üretmek için kullanılan gerçek trafik kayıtları gibi bilgiler tutulur. Raporlama modülü ile STİ üretimi için kullanılan raporlama fonksiyonları barındırılır. Saldırı tespit sistemi modülü ile ise trafik kayıtları üzerinde incelemeler gerçekleştirmek için kullanılan STS işlevleri yerine getirilir. İlgili STS, veri madenciliğinden faydalanılarak geliştirilmiştir. Bu kapsamda veri madenciliği adımları bu modül üzerinde gerçekleştirilir. Ağ trafiği dinleme modülü ile canlı trafik bilgileri yakalanmakta veya kaydedilmiş trafik bilgileri okunmaktadır. Belirli bir formatta elde edilen bu bilgiler, saldırı tespit sistemi modülü tarafından işlenmek üzere alınmaktadır. Ham veri üretmek amacıyla saldırı yüzeyi oluşturmak için zafiyet simülasyon modülü ve saldırı kayıtları üretmek için tehdit simülasyon modülü kullanılır. Zafiyet simülasyon modülü ve tehdit simülasyon modülü ile sisteme, hedeflenmiş saldırılar için STİ oluşturma yetkinliği kazandırılır.

STS için saldırı tiplerinin öğrenilebilmesi adına kullanılması muhtemel çeşitli veri kümeleri incelenmiştir. Yapılan incelemeler neticesinde KDD CUP 99 ve NSL-KDD veri kümeleri tercih edilmiştir. Önerilen sistemin uygulaması geliştirilerek genel kabul görmüş KDD CUP 99 veri kümesi üzerinden veri madenciliği algoritmalarının başarımları karşılaştırılmıştır. Buna ek olarak, önerilen sistem üzerinde uygulanan benzer adımlar veri madenciliğinde güvenilirliğini kanıtlamış Weka aracı üzerinde de gerçekleştirilmiş ve elde edilen başarımların önerilen sistemin başarımlarına benzer olduğu gösterilmiştir. Saldırı tiplerini öğrenmek ve trafik kayıtlarını sınıflandırmak için veri madenciliği tekniği olarak

karar ağacı algoritmalarından CART ve ID3 tercih edilmiştir. Yapılan karşılaştırmalar için %10 KDD CUP 99 veri kümesi kullanılmış ve tabakalı 10 katlamalı çapraz doğrulama uygulanmıştır. İlgili sonuçlar ayrıca literatürde yer alan benzer çalışmalar ile de kıyaslanmıştır. Önerilen sistem üzerinde gerçekleştirilen tabakalı 10 katlamalı çapraz doğrulama için %10 KDD CUP 99 veri kümesi üzerinde sırası ile CART ve ID3 algoritmaları için başarımların ortalaması %99,91 ile %99,4 olarak gözlemlenmiştir.

KDD CUP 99 ve %10 KDD CUP 99 veri kümelerinin birçok dezavantajı bulunduğundan dolayı NSL-KDD veri kümesi, STİ üretmek için ihtiyaç duyulan saldırı tiplerini öğrenmek için kullanılmıştır. Saldırı tipleri NSL-KDD Eğitim veri kümesinin tamamı kullanılarak öğrenilmiştir. Bu sebeple NSL-KDD Eğitim veri kümesi üzerinde çalıştırılan veri madenciliği algoritmalarının karşılaştırılması için bu veri kümesi üzerinde tabakalı 10 katlamalı çapraz doğrulama kullanılmış ve literatürde yer alan benzer çalışmalar ile kıyaslanmıştır. Önerilen sistem üzerinde gerçekleştirilen tabakalı 10 katlamalı çapraz doğrulama için NSL-KDD veri kümesi üzerinde sırası ile CART ve ID3 algoritmaları için başarımların ortalaması %98,76 ile %96,18 olarak ölçülmüştür.

Önerilen sistemin uygulaması için örnek bir çalışma olarak 24 saat boyunca internete açık çalıştırılan bir sunucu üzerinde bal küpü sistemleri ile saldırganların dikkati çekilerek ilgili trafik kayıtları kaydedilmiştir. Kaydedilmiş trafik bilgisi, dinleme modülü üzerine yerleştirilerek önerilen sistemin ilgili kayıtları sınıflandırarak STİ oluşturması sağlanmıştır. Önerilen sistemin STİ üretmek için başarılı bir şekilde kolaylıkla kullanılabileceği gösterilmiş ve internet ortamına açılacak olan sunucular için kullanılabilecek STİ üretilmiştir.

Gerçekleştirilen bu araştırmanın bildiğimiz kadarı ile literatürdeki benzer çalışmalardan farkı; KDD CUP 99 ve NSL-KDD veri kümesi formatında canlı veya kaydedilmiş gerçek trafik kayıtları üzerinde analizler gerçekleştirebilecek yetkinlikte bir STS geliştirilmesi ve STİ oluşturmak için gerçek trafik kayıtları üzerinde incelemeler yapan veri madenciliği temelli bir STS'den faydalanılmış olmasıdır.

Öğrenilmiş ve siber tehdit istihbaratı üretilmiş saldırılar, NSL-KDD veri kümesi içinde bulunan saldırı tipleri ile sınırlıdır. Buna istinaden, geliştirilmiş yazılım benzer veri kümelerine adapte olabilecek yetkinliktedir. Ayar dosyasının kullanılacak veri kümesine göre özelleştirilmesi ile farklı saldırı tipleri barındıran benzer veri kümeleri de

kullanılabilir. Ayrıca sistem, uzman tarafından bilinmeyen saldırıların tanımlanmasına imkan sağlayarak öğrenme kümesini güncelleyebilmektedir. Tüm bunlara ek olarak, sistem tarafından üretilen STİ'nin kapsamı yalnızca ağ trafiğinden oluşturulabilir teknik STİ ile sınırlıdır. Bu sebeple STIX formatında üretilen teknik STİ; yalnızca saldırı düzeni (attack pattern), gösterge (indicator) ve ilişki (relationship) nesnelerinden oluşturulabilmektedir. Ayrıca, saldırganların dikkatini çekmek için kullanılan bal küpü sistemleri, bazı tarama araçları vasıtası ile fark edilebilir [29]. Bir sistemin bal küpü olduğunu anlayan kötü niyetli kullanıcı saldırıyı ilerletmeyebilir. Bu durum kötü niyetli kullanıcının yapacağı asıl saldırı için bilgi toplamayı önleyebilmektedir. Bu kısıtlara ek olarak internete açık olarak çalıştırılan sunucu için alınan bulut hizmeti, bazı saldırı trafiğini ve/veya saldırıları engelleyebilmektedir. Bu sebepten dolayı bazı saldırganlar ve saldırıları hakkında bilgi toplanamayabilir.

2. TEMEL BİLGİLER, KAVRAMLAR VE TERİMLER

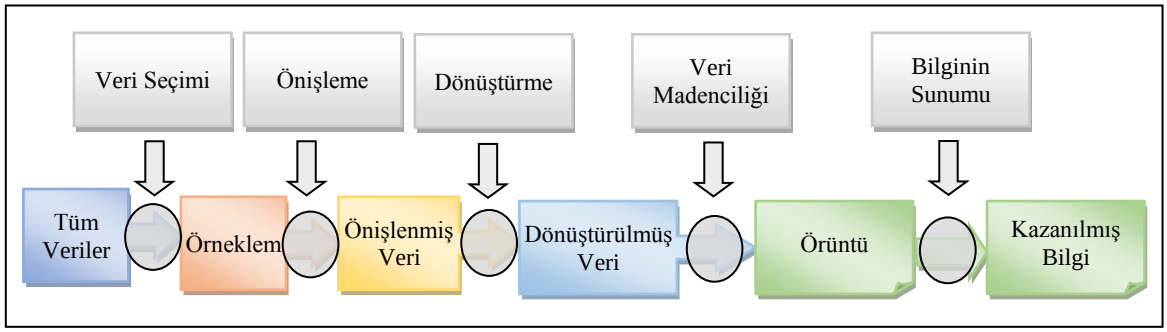
2.1. Veri Madenciliği

Veri madenciliği, büyük miktardaki veri üzerinde gerçekleştirilecek analizler ile eğilimlerin, örüntülerin ve anlamlı ilişkilerin keşfedilme sürecidir [30]. Ancak disiplinler arası bir konu olduğundan birçok tanımı mevcuttur. Tam da bu sebepten bilgi çıkarımı, veri analizi, örüntü analizi, veri arkeolojisi ve veri tarama gibi birçok isim ile kullanılmaktadır [31].

Veri madenciliğinde asıl amacın veriden veri elde etmek değil büyük miktardaki veriden desen ve bilgi çıkarmak olduğundan literatür taramasında veri madenciliği teriminin yanlış bir isimlendirme olarak kabul edildiği görülmüştür [31]. Dolayısı ile veri madenciliği yerine veri tabanından bilgi keşfi terimi sıklıkla tercih edilmektedir [32]. Ayrıca literatür taramasında görülmüştür ki veri tabanından bilgi keşfi veriden bilginin elde edilmesi için ele alınan tüm süreçleri ifade ederken veri madenciliği bu süreçlerden yalnızca biri olarak kabul edilmektedir [33, 34].

2.1.1. Veri tabanından bilgi keşfi süreçleri

Veri tabanından bilgi keşfi yinelemeli bir süreçtir [35]. Bu süreç veri seçimi, önişleme, dönüştürme, veri madenciliği ve bilginin sunumu gibi 5 adımdan oluşmaktadır [36]. Tüm adımlar Şekil 2.1 üzerinde görülebilir.



Şekil 2.1. Veri tabanından bilgi keşfi süreçleri

İlk olarak veri tabanından bilgi keşfi için hazırlık yapılmalıdır. Bu aşamada üzerinde çalışılacak alan ile ilgili ön bilgi edinilir [37]. Ayrıca ortaya çıkarılması amaçlanan bilginin son kullanıcı açısından amacının belirlenmesini de içermektedir.

Veri seçimi adımı ile bilgi keşfi için kullanılacak veriler belirlenmelidir [38]. Toplanmış farklı veriler bu adımda birleştirilerek bir veri kümesi oluşturulabilir [37]. Oluşturulan bu veri kümesinden bilgi keşfi için kullanılmak istenilen veriler ayrıştırılır [32]. Bu sayede belirlenmiş amaca ulaşmak için kullanılabilecek yararlı veriler seçilmiş ve artık veri temizlenmiş olur. Böylece ilgisiz ve alakasız veriler üzerinde işlem yapmaktan kaçınılmaktadır. Dolayısı ile daha doğru sonuçlar elde edilecektir. Artık veriden dolayı karşılaşılabilecek problemler bir sonraki başlık olan "veri madenciliğinde dikkat edilmesi gereken hususlar" altında incelenmiştir. Veri seçimi adımının sonunda örneklem elde edilir. Bu yüzden veri seçme adımı örneklem çıkarma adımı olarak da kabul edilebilir.

Örneklem çıkarma işleminde olasılıklı örnekleme ve olasılıksız örnekleme olmak üzere iki metot kullanılabilir [39]. Olasılıklı örneklemeye basit rastlantısal örnekleme ve katmanlı rastlantılı örnekleme teknikleri, olasılıksız örneklemeye ise yargısal örnekleme örnek olarak gösterilebilir [39-41]. Yargısal örnekleme insan seçimine dayalı olduğundan istatistiksel ve matematiksel olarak ifade edilemez. Yargısal örnekleme de, araştırmacı verileri sahip olduğu bilgisine güvenerek bizzat kendisi seçer [42]. Araştırmacının önyargılı fikirleri ile hareket edildiğinden sonuçlar tüm veri için doğru olmayabilir çünkü seçilmiş örneklem geri kalan veriyi tam anlamıyla temsil edemeyebilir. Pilot çalışmalarda bu yöntem tercih edilebilir. Olasılıklı örneklemede ise istatistiksel ve matematiksel yöntemler temel alınır. Basit rastlantısal örneklemede tüm kayıtlar seçilebilmek için aynı olasılığa sahiptir [43]. Bu aynı olasılıktaki her kayıttan rastgele olarak istenilen sayıda seçilir. Bu örnekleme yöntemi ile oluşturulan örneklem de geri kalan veriyi tam olarak temsil edemediğinden çıkarılmış bilgi tüm veri için doğru olmayabilir. Sıklıkla kullanılan katmanlı rastgele örneklemede ise örneklemin geri kalan veriyi aynen temsil edebilmesi amaçlanarak veride bulunan sınıfların oranı ile örneklemde bulunan sınıfların oranı eşit tutulur [44]. Her bir sınıftan sınıf sayısı ile doğru orantılı olacak şekilde kayıt örneklem kümesine aktarılır. Sınıf içindeki her kaydın seçilme olasılığı burada da eşittir.

Önişleme adımı ile tutarlı veri elde etmek amaçlanmaktadır [36]. Eksik değerler saptanarak bu uygunsuzluğun nasıl ele alınacağına karar verilir [37]. İlgili veri kümesi analiz edilerek gürültü olduğu saptanan veriler temizlenir [32]. Ayrıca bu aşamada yapılacak sapan veri incelemesi ile veri kümesinin gerisinden farklı olan ve diğerlerinden bariz bir şekilde ayrılan verilerin temizlenmesi işlemi de ele alınır [34]. Tüm bu işlemler veri temizleme olarak kabul edilebilir [35].

Dönüştürme adımında öznitelik seçme, öznitelik oluşturma ve öznitelik dönüştürme işlemleri ele alınmaktadır [32, 37]. Sonuca ulaşmada etkisi olmayan özniteliklerin çıkarılarak veri madenciliği algoritması için yalnızca belirli özniteliklerin seçilmesi ile hem veri madenciliği algoritması gereksiz bir yükten kurtarılmakta hem de daha doğru sonuçlar elde edilmektedir. Örnek olarak insanların spor alışkanlıklarının analizinde kullanılacak veri madenciliği algoritmasının sonuca ulaşmasında bir etkisi olmayacak telefon numarası özniteliğinin kullanılmayarak silinmesi gösterilebilir. Öznitelik oluşturma ile gerekli görülen öznitelikler birleştirilebilir ya da mevcut özniteliklerden farklı bir öznitelik hesaplanabilir. Bu şekilde verinin boyutu artırılabilmektedir. Öznitelik oluşturma işlevinin doğru bir şekilde kullanılması ile veri madenciliği algoritmasının başarısı yükseltilebilmekte ancak gereksiz şekilde kullanıldığında ise ya hiçbir etki göstermemekte ya da algoritmanın başarısını düşürmektedir. Ek olarak öznitelik dönüştürme ile örneklem kümesi kayıtları üzerinde veri madenciliği algoritması için değişikler yapılabilir. Veri üzerinde yapılacak normalizasyon işlemleri ya da sürekli verilerin kesikli verilere dönüştürülmesi bu değişikliklere örnek olarak verilebilir. Yapılabilecek normalizasyon işlemlerine veri normalizasyonu tekniklerinden min-max normalizasyonu ile kayıtların ilgili öznitelik değerlerine 0 ile 1 arasında değerler verilmesi örnek olarak gösterilebilir.

Veri seçimi, önişleme ve dönüştürme adımlarının ardından ilgili verinin, veri madenciliği adımları için hazır duruma geldiği öngörülebilir. Veri madenciliği adımları olarak kullanılacak veri madenciliği metodunun seçilmesi [32], seçilen metot için ilgili algoritmanın belirlenmesi [34] ve bu algoritmanın uygulanarak örüntünün elde edilmesi [37] sıralanabilir.

Veri tabanından bilgi keşfi sürecinin son adımı olarak elde edilen örüntü görselleştirme teknikleri ile bilgiye çevrilerek kullanıcıya sunulur. Dolayısı ile bu adım bilginin sunumu adımı olarak tanımlanabilir.

Veri tabanından bilgi keşfi sürecinde her adım arasında yinelemeli olarak döngüler görülebilir [32]. Her adımdan bir veya daha önceki adıma dönülerek farklı bir yöntem, teknik veya yaklaşım denenebilir. Bu deneme yanılma ile daha iyi bir sonuç elde edilmeye çalışılabilir.

2.1.2. Veri madenciliğinde dikkat edilmesi gereken hususlar

Veri tabanından bilgi keşfi sürecinin ve veri madenciliği algoritma ve tekniklerinin oldukça güçlü bir şekilde tasarlanmış olmasına karşın uygulanması sırasında birçok zorlukla karşılaşılmaktadır. Karşılaşılan zorluklar için geliştirilmiş uygulamanın performansının zayıf olması, verinin istenilen kalitede olmaması veya kullanılan metot ve tekniklerin uygunsuz olması örnek olarak gösterilebilir. Bahsi geçen bu ve benzeri problemler doğru bir veri tabanından bilgi keşfi sürecinin uygulanması ile aşılabilir. İşte tam da bu noktada ilgili süreç başarı ile işletilmiş olur.

Veri madenciliğinin uygulanabilmesi adına öncelikle ilgili uygulama için veri toplanmalıdır. Ancak toplanan veri her zaman istenilen kalitede olmayabilir. İçerisinde gürültülü ve eksik veriler barındırabilir [45]. Bu durum ilgili veri üzerinde uygulanacak olan veri madenciliği algoritmalarının başarılı bir şekilde çalışmasını engellemektedir. Gürültü, veri girişi sırasında oluşabileceği gibi ilgili değerlerin işlenmesi ya da taşınması sırasında da meydana gelebilmektedir. Gürültülü değerler hatalı değerlerdir [46]. Dolayısı ile veri madenciliği algoritmasının sonucunu yanıltabilir. Bu iki duruma ek olarak artık veride veri madenciliği uygulamasının performansını düşürmektedir. Artık veri terimi amaçlanan bilgi çıkarımı için faydası olmayan veriler için kullanılmaktadır [47]. Artık veriler, istenilen sonuca ulaşmada kullanılacak veri madenciliği algoritmaları için gereksiz nitelikteki verilerdir. Dolayısı ile bu tip artık verilerin ve gürültülü verilerin veri kümesinden çıkarılması gerekmektedir. Ancak eksik veri için farklı yöntemler tercih edilebilir [48]. Eksik veri içeren kayıtların veri kümesinden çıkarılabileceği gibi ilgili özniteliğin ortalaması bu eksik verinin yerine kullanılabilir. Bu sayede ilgili değerler doldurulmuş olur. Bu çözüme ek olarak daha önceden belirlenmiş sabit bir değerde eksik verilerin doldurulması için kullanılabilir. Artık verilerin ve gürültülü verilerin ilgili veri kümesinde bulunması durumunda veri madenciliği algoritmalarının performansını düşürebilir ya da yanlış sonuçlar üretmesine sebep verebilir. Ancak eksik veri olması durumunda algoritma çalışmasını durdurabilir. Çünkü boş veri, herhangi bir veriye eşit değildir [48]. Herhangi bir veriye eşit olmadığı gibi kendine de eşit değildir. Bu yüzden boş değerler veri madenciliği algoritmaları açısından büyük problem teşkil etmektedir.

Günümüz dünyasında birçok kurum, kuruluş veya kişiler sahip oldukları verilerin büyüklüğünden dolayı tek bir veri tabanı sunucusu ile yetinememektedir. Bu yüzden dağıtık sunucu mimarisi kullanmakta ve hatta farklı veri tabanı sistemleri ile

çalışmaktadırlar. Bu durum veri madenciliği uygulamasının dağıtık ortamdaki veri ile çalışmasını ya da ilgili verilerin tek bir noktada toplanmasını gerektirmektedir. Farklı veri tabanı sistemlerinde bulunan verilerin bir başka veri tabanı sistemine aktarılması oldukça zor bir iştir. Buna ek olarak verinin çok büyük olması gibi bazı durumlarda tek bir noktada tüm veriyi toplamak imkansız olabilir. Bu yüzden veri madenciliği uygulamalarının ve algoritmalarının dağıtık veri tabanı sistemleri ile çalışabilmesi için geliştirilmesi gerekebilmektedir [49]. Çünkü birçok veri madenciliği uygulaması dağıtık veri tabanı sistemleri ile çalışmaya olanak sağlamamaktadır.

Birçok insan için veri tabanı denilince akla ilk olarak ilişkisel veri tabanı veri modeli olan ilişkisel veri modeli gelir. Bu model 1969 yılında İngiliz bilgisayar bilimcisi Edgar F. Codd tarafından oluşturulmuştur [50]. Kayıtların tablo üzerinde gösterildiği bu veri modelinde değerler genellikle metin veya sayılardan oluşmaktadır. İnsan tarafından dahi okumanın ve anlamanın oldukça basit olduğu bu model veri madenciliği uygulamalarında sıklıkla kullanılmaktadır. Veri modelinin basit olmasından dolayı veri madenciliği algoritmaları ile kolaylıkla çalışılmaktadır. Fakat veri tabanından bilgi keşfi sürecinde yalnızca metin veya sayı tabanlı veriler kullanılmamaktadır. Karmaşık olan video, ses ve resim gibi çoklu ortam verileri üzerinde de veri madenciliği teknikleri ile çalışılmaktadır [51, 52]. Bu verilerin temsilinin ve anlamlandırmasının karmaşıklığı ile birçok veri madenciliği adımının uygulamasının kolay olmamasından dolayı veri madenciliği teknikleri ile bu tip veriler üzerinde çalışmak oldukça zorlaşabilmektedir [53]. Bu gibi karmaşık veriler ile çalışmak için yeni araç ve metotlara ihtiyaç duyulabilmektedir.

Veri tabanından bilgi keşfi sürecinin performansı veri madenciliğinde kullanılacak olan algoritma ve tekniklerinin verimliliğine doğrudan bağlıdır. Dolayısı ile veri madenciliğinde kullanılmak üzere seçilmiş olan metot ve algoritma, sonuca ulaşmada oldukça büyük öneme sahiptir [54]. Yanlış seçilmiş metot veya algoritma istenilen sonuca ulaşmadaki en büyük engeldir. Doğru veri madenciliği yöntemi kullanmak için literatür taraması yapılarak benzer çalışmalarda kullanılmış algoritmalar tercih edilebilir veya uygulanacak farklı algoritmaların sonuçları karşılaştırılarak biri üzerinde karar kılınabilir.

Veri madenciliği sonucunda elde edilen bilginin kullanıcıya gösterimi ve sunulması işlemi oldukça zorlaşabilmektedir. Girdi olarak alınacak kazanılan bilginin ve çıktı olarak verilecek istenilen sunumun oldukça karmaşık olduğu durumlarda veri görselleştirme tekniklerinden faydalanılmalıdır [31].

Veri tabanından bilgi keşfi sürecinde kullanılacak olan veri, ulaşılmak istenilen sonuç için yeterli sayıda öznitelik barındırmıyor olabilir. Bu tür veriler sınırlı bilgi olarak adlandırılabilir [54]. Sınırlı veriden dolayı kazanılması amaçlanan bilgi hiç elde edilemeyebilir ya da oldukça düşük bir doğruluk ile elde edilebilir.

2.1.3. Veri madenciliğinde kullanılan modeller ve algoritmalar

Veri madenciliği metotları tanımlayıcı ve tahmin edici modeller olmak üzere iki ana başlık altında incelenebilir [55]. Sınıflandırma ve regresyon tahmin edici model olmasına karşın kümeleme ve birliktelik kuralları tanımlayıcı modellerdir [54]. Tüm bu bilgilere ek olarak tahmin edici modeller kestirime dayalı modeller olarak da bilinmektedir. Ayrıca denetimli öğrenme ve denetimsiz öğrenme olmak üzere veri madenciliği modelleri iki sınıfa da ayrılabilir [56]. Denetimli öğrenmede öğrenme kümesindeki verilerin sınıf etiketlerine sahip olması beklenmektedir. Sınıfı bilinen kayıtlardan ilgili sınıf için karakteristik özellikler çıkarılarak test verisinin sınıfı tahmin edilir. Bu öğrenme metoduna sınıflandırma ve regresyon analizi örnek olarak gösterilebilir. Denetimsiz öğrenmede ise öğrenme kümesindeki kayıtların sınıf özniteliğine sahip olması gerekli değildir. Çünkü kayıtların hangi sınıfa ait olduğu öğrenme aşamasında dikkate alınmaz ve sınıflar algoritma tarafından oluşturulur. Bu öğrenme metoduna örnek olarak kümeleme ve birliktelik kuralları verilebilir [54].

Veri madenciliğinde kullanılacak olan veri kaynakları iki şekilde gruplandırılabilir. Belirli örüntülerin çıkarılması için öğrenme kümesi olarak kullanılan kayıtlara eğitim seti, çıkarılan örüntülerin test edilmesi için kullanılan kayıtlara ise test seti denir.

Sınıflandırma ve regresyon

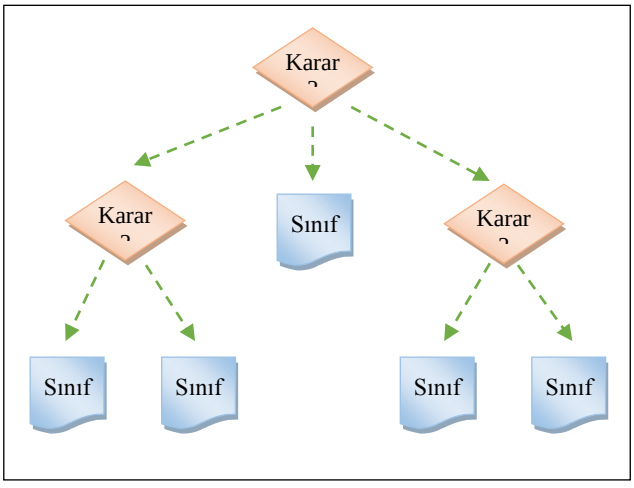
Sınıflandırma ve regresyon, eğitim veri kümesi olarak kullanılan örneklemden öğrenilen sınıflara göre ilgili verilerin sınıf etiketlerini tahmin etme işlemidir. Sınıflandırma ile kategorik sınıf etiketleri tahmin edilirken regresyon ile sürekli değerden oluşan sayısal sınıf etiketleri tahmin edilir [57, 58]. Örneğin belirli kriterler göz önünde bulundurularak bir mahalledeki evlerin yürünebilirlik açısından kategorisinin belirlenmesi için sınıflandırma tercih edilebilirken regresyon ise ilgili evlerin mali değerlerini tahmin etmek için kullanılabilir [59].

Regresyon için girdiler bağımsız değişken, çıktılar ise bağımlı değişken olarak adlandırılır[60]. Regresyon ile bir veya daha fazla bağımsız değişken ile bağımlı değişkeni elde edebilecek model üretmek amaçlanır. Tek bağımsız değişken ile bir bağımlı değişken arasındaki ilişkinin analizine tek değişkenli regresyon, birden çok bağımsız değişken ile bir bağımlı değişken arasındaki ilişkinin incelenmesine ise çok değişkenli regresyon adı verilir [61]. Regresyon modellemesi trend analizi, finansal tahmin, zaman serisi tahmini, biyomedikal ve ilaç yanıtı modellemesi için kullanılabilir [59].

Sınıflandırma ile kayıtlardan oluşan veri, sınıf kümesine dönüştürülmektedir. Sınıflandırmanın kullanıldığı bazı veri madenciliği uygulamalarına konuşma tanıma, el yazısı tanıma, belge sınıflandırması gibi örnekler verilebilir [62-64]. Sınıflandırma algoritmalarından bazıları karar ağaçları, yapay sinir ağları, k-en yakın komşu algoritması, naive bayes sınıflandırıcı algoritması ve destek vektör makineleridir [65].

Karar ağaçları

Karar ağaçları kolaylıkla yorumlanabilmesinden ve anlaşılabilirliğinin basitliğinden dolayı sınıflandırma için yaygın olarak kullanılmaktadır. Hem kategorik hem de nümerik veriler ile kullanılabilir [66]. Karar ağacında veriler sürekli olarak belirlenen bir noktaya göre bölünür. Böylece her bölünme noktasının bulunması ile bir karar düğümü oluşmuş olur. Veri en sonunda bölünemeyecek duruma geldiğinde geriye kalan sınıf değerlerinden nihai sonuç belirlenir. Böylece karar düğümlerinin sonunda ilgili sınıf tahmini barındıran yapraklar oluşturulmuş olur. Şekil 2.2 üzerinden görülebileceği üzere oluşturulan karar düğümleri üzerinden adım adım tahmin edilen sınıf değerine ulaşılmaktadır.



Şekil 2.2. Karar ağacında bulunan karar düğümleri ve sınıf yaprakları

Karar ağacı oluşturmak için çalışılan alan ile ilgili ön bilgi edinmeye olan ihtiyaç oldukça düşüktür [67]. Dolayısı ile kullanıcılar tecrübe sahibi olmadığı konular hakkında dahi sınıflandırma yapabilecek bir karar ağacı oluşturabilirler. Karar ağacı oluşturmak için kullanılan algoritmalarına örnek olarak ID3, C4.5, C5 ve CART gösterilebilir [68]. Tüm bu algoritmalar verinin bölüneceği noktanın bulunmasında kullanılan tekniğin farklılığından dolayı birbirinden ayrılır. Bu tekniklere bilgi kazanımı ve gini indeksi örnek olarak verilebilir [69]. Tüm bu tekniklerin düşük hesaplama karmaşıklığına sahip olmasından dolayı karar ağaçlarının oluşması için birçok veri madenciliği algoritmasına göre büyük işlem yüküne ihtiyaç duyulmaz.

ID3 algoritması

ID3, Ross Quinlan tarafından geliştirilmiş basit bir karar ağacı algoritmasıdır [70]. Entropi ve bilgi kazancı değerlerinin hesaplanması ile ağaç yukarıdan aşağıya doğru oluşturulur. Entropi belirsizliğin ölçüsüdür[71]. Tüm örnekler aynı sınıfa ait ise belirsizlik 0'dır. Bunun aksine örneklerin sınıfları eşit sayıda ise belirsizlik 1'dir.

Entropi kullanılarak bilgi kazancı değeri hesaplanır. Hesaplanan bilgi kazancı değerlerinden en büyüğü ağaç oluşturmak için seçilir [70, 72]. Hesaplamada kullanılacak olan entropi değeri Eş. 2.1 üzerinde gösterilmiştir [71] [72].

$$E(S) = -\sum_{i=1}^{n} f(i) \log_2 f(i) \tag{2.1}$$

E(S) ile ilgili adım için S veri kümesinin entropi değeri ifade edilmektedir. n değeri ise S veri kümesinde bulunan farklı sınıfların sayısıdır. f(i), S veri kümesinde bulunan i sınıfı için hesaplanacak olan frekans değeridir.

ID3 algoritmasında ilgili sınıf etiketine ulaşmak için kullanılabilecek en iyi özniteliğin seçilmesi adına bilgi kazancından faydalanılır [73]. İlgili hesaplama Eş. 2.2 üzerinde gösterilmiştir [71].

$$G(S, A) = E(S) - \sum_{i=1}^{m} f(A_i) E(S_{A_i}) \tag{2.2}$$

G(S, A) ile S veri kümesinin A özniteliğine göre bölündüğünde elde edilecek olan bilgi kazancı ifade edilmektedir. m değeri S veri kümesinde bulunan A özniteliğinin barındırdığı farklı değerlerin sayısıdır. f(A_i) ile S veri kümesinde bulunan A özniteliğinin A_i değerinin frekansı temsil edilmektedir. A_i ise A özniteliğinin farklı değerlerinden i'inci olanını ifade

eder. S_{A_i}, S veri kümesinde bulunan A özniteliğinin A_i değerine sahip olan tüm kayıtların kümesini belirtir.

Veri kümesinde bulunan tüm öznitelikler için bilgi kazancı hesaplanır. En büyük bilgi kazancına sahip öznitelik seçilir [70, 72]. Böylece ağacın kök elamanı seçilmiş olur. Seçilen özniteliğin her bir değerine göre veri alt kümelere bölünür [74]. Bölünen her bir alt küme üzerinde de ilgili hesaplamalar yapılarak en verimli özniteliğin seçilmesi sağlanır. En verimli özniteliğe göre ilgili alt küme tekrar bölünür. Bu şekilde yinelemeli olarak hesaplamalar ve bölmeler ile devam edilir. Böylece ağaç adım adım yukarıdan aşağıya doğru oluşturulur.

CART algoritması

CART; Leo Breiman, Jerome Friedman, Richard Olshen ve Charles Stone tarafından geliştirilmiş bir karar ağacı algoritmasıdır [75]. Tüm öznitelikler yalnızca ikiye bölünebilirdir. Dolayısı ile CART algoritması ile ikili ağaçlar oluşturulur [76]. ID3 algoritmasında olduğu gibi CART algoritmasında da ağaç yukarıdan aşağıya doğru adım adım oluşturulmaktadır.

Gini index değeri ve kirlilikte azalma değeri hesaplanarak en uygun öznitelik ve ayırma noktası belirlenir [75, 77]. Belirlenen ayırma noktasına göre veri alt kümelere bölünerek her alt küme için kirlilikte azalma değerleri hesaplanır ve en uygun özniteliğin seçilmesi ile yinelemeli olarak devam edilir [78]. Eş. 2.3 üzerinde gini index hesaplaması gösterilmiştir. [79, 80]

$$\text{gini(S)} = 1 - \sum_{i=1}^{n} p(i)^2 \tag{2.3}$$

gini(S) ile S veri kümesine ait gini index değeri hesaplanmıştır. n değeri S veri kümesinde bulunan farklı sınıf sayısıdır. p(i) ise ilgili sınıfın görülme sıklığıdır.

Gini index değeri kullanılarak her öznitelik için kirlilikte azalma değeri hesaplanır. Hesaplanan bu kirlilikte azalma değerlerinden en büyüğüne sahip olan öznitelik seçilir [75]. Eş. 2.4 üzerinde kirlilikte azalma değerinin hesaplanması gösterilmiştir. [75] [79 – 84]

$$\Delta\text{I(A)} = \text{gini(S)} - gini_A(S) \tag{2.4}$$

ΔI(A) ile kirlilikte azalma değeri ifade edilir. S veri kümesinin A özniteliğine göre bölündüğü durum için gini index değeri olan $gini_A(S)$ hesaplaması Eş. 2.5 üzerinde gösterilmiştir. [75] [79 – 84]

$$gini_A(S) = \frac{|S_1|}{|S|} gini(S_1) + \frac{|S_2|}{|S|} gini(S_2) \quad (2.5)$$

$gini_A(S)$ değeri hesaplanırken ilgili S veri kümesinde bulunan A özniteliğinin tüm farklı değerleri için $gini_A(S)$ hesaplanır. En küçük $gini_A(S)$ değerine sahip olan A özniteliği değeri ayırma noktası olarak kabul edilir. Bu ayırma noktası ile hesaplanmış $gini_A(S)$ değeri ilgili özniteliğin ΔI(A) değerini hesaplamak için kullanılır.

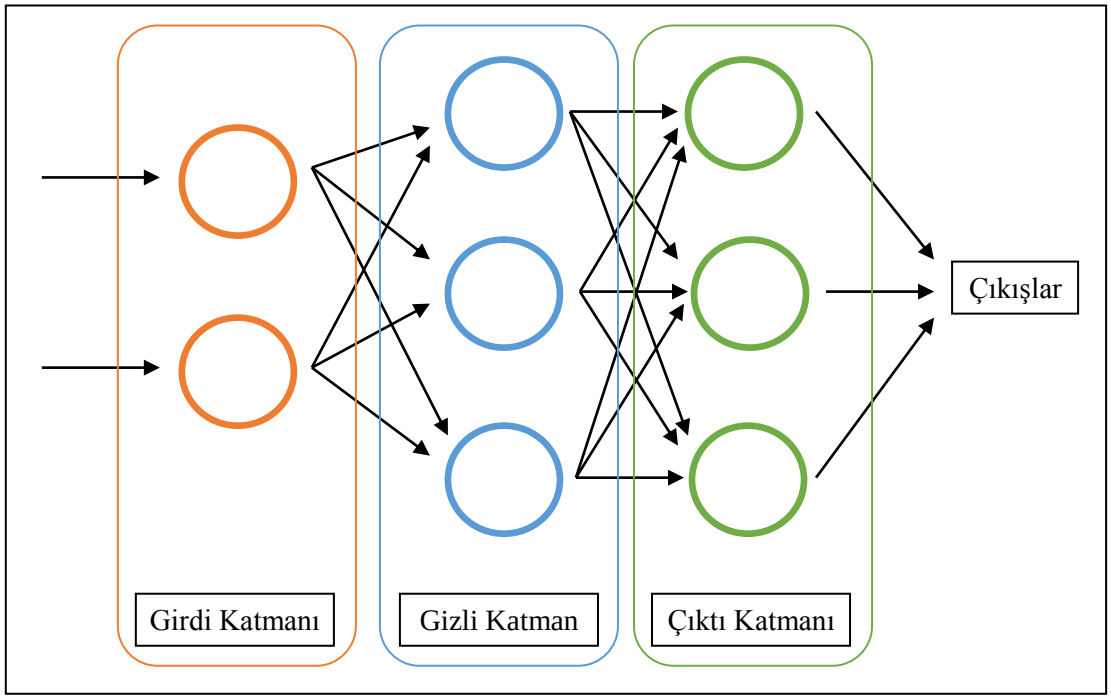
Tüm hesaplamalar sonucunda ΔI(A) değeri en büyük olan öznitelik seçilir ve ilgili özniteliğin $gini_A(S)$ değerini en küçük yapan öznitelik değeri bölünme noktası olarak kabul edilir. Böylece kullanılacak olan öznitelik ve bölünme noktası bulunmuş olur. Sonuç olarak, ID3 algoritmasında olduğu gibi yinelemeli olarak veri kümesi alt kümelere bölünerek ilgili hesaplamalar ile en verimli öznitelikler seçilir. Böylece adım adım karar ağacı oluşturulur.

Yapay sinir ağları

Yapay sinir ağlarında, canlıların sahip olduğu biyolojik sinir ağlarından esinlenilmiştir [85]. Genellikle üç katmandan oluşur. Bunlar girdi katmanı, gizli katman ve çıktı katmanıdır [86]. Ara katman olarak da bilinen gizli katman bazen kullanılmamaktadır. Böyle bir durumda yalnızca girdi katmanı ve çıktı katmanı bulunur. Bu mimariye tek katmanlı mimari denir [87]. Genellikle doğrusal problemlerin çözümü için tercih edilir [88]. Şekil 2.3 üzerinde katmanlar görülebilir. Ayrıca ilgili şekil üzerinden bir yapay sinir ağı hücresi olan nöronlarda görülebilmektedir.

Her nöron bir önceki nörondan girdi alır. Alınan bu girdiler aktivasyon fonksiyonuna tabi tutularak çıktı üretilir. Aktivasyon fonksiyonları için adımsal aktivasyon fonksiyonu, sigmoid aktivasyon fonksiyonu, gaussian aktivasyon fonksiyonu örnek olarak gösterilebilir [89, 90]. Üretilen çıktı bir sonraki nörona girdi olarak verilir. Yapay sinir ağları modelleri ileri beslemeli ve geri beslemeli olmak üzere ikiye ayrılır [91]. İleri beslemeli modellerde bilgi akışı tek yönlü olarak ileri doğrudur. Her nöron kendinden bir sonraki katmanda bulunan nöronlara çıktısını girdi olarak verir. Geri beslemeli modellerde ise bilgi akışı çift yönlü olarak hem ileri hem de geri doğru olabilir. Dolayısı ile bir nöron çıktısını bir

sonraki katmanda bulunan nöronlara ve/veya bir önceki nöronlara girdi olarak verebilir. Bu tip modeller zaman serilerinin tahmininde oldukça büyük başarı göstermektedir [92]. Başarılı örnekler olarak Jordan ve Hopfield ağları gösterilebilir [93]. İleri yönlü veya geri yönlü olarak taşınan bu girdi değerleri ilgili nöron için ağırlık kullanılarak alınır. Bir nöronun her girdisi için bir ağırlık değeri vardır. Yapay sinir ağları üzerinde bilgi bu ağırlıkların değerleridir. Dolayısı ile yapay sinir ağlarının öğrenme işlemi ağırlıkların belirlenmesinden ibarettir. En iyi ağırlık değerlerinin bulunması ile öğrenme tamamlanmaktadır. Yapay sinir ağlarında ağırlıklar deneme yanılma yöntemi ile bulunduğundan öğrenme süresi önceden kestirilemez.



Şekil 2.3. Yapay sinir ağları katmanları

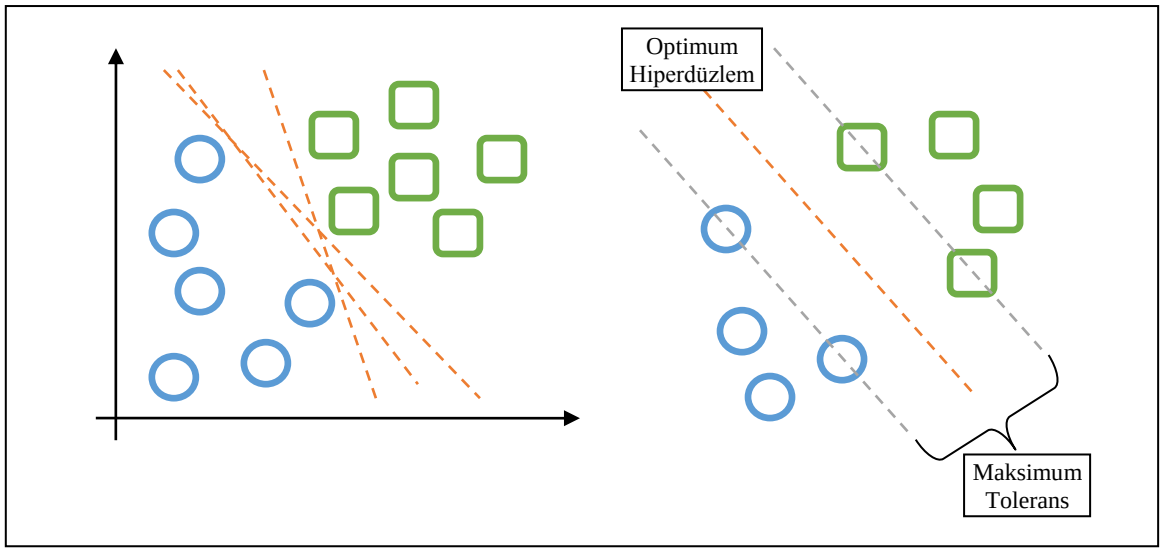
K-en yakın komşu algoritması

K-en yakın komşu algoritması sınıflandırma algoritmalarının kullanımında gürültülü değerler içeren verilere rağmen doğru sonuçlar üretebildiğinden sıklıkla tercih edilmektedir [94]. K değeri kontrol edilecek eleman sayısını temsil etmektedir. Eğitim süreci k-en yakın komşu algoritmasında yoktur. Sınıf değeri tahmin edilecek veri, k adet komşusu ile karşılaştırılarak en yakın bulunan komşunun sınıfına dahil edilir. Bu aşamada uzaklık kontrolleri için Öklid uzaklık fonksiyonu ve Manhattan uzaklık fonksiyonu sıklıkla kullanılmaktadır [95]. Öklid uzaklık fonksiyonu iki nokta arasındaki uzaklığı Öklid teorisi ile hesaplamaktadır. Manhattan uzaklığı ise iki nokta arasındaki uzaklığı her boyutun

koordinatları arasındaki farkların mutlak değerlerinin toplanması ile bulunur. Bu uzaklık fonksiyonlarına ek olarak Minkowski uzaklık fonksiyonu ve Chebyschev uzaklık fonksiyonu da tercih edilebilir [96]. Tüm bu uzaklık fonksiyonları değerlerin nümerik olduğu durumlarda tercih edilebilmektedir. Ancak değerler kategorik verilerden oluşuyorsa Hamming mesafesi kullanılabilir [97]. K-en yakın komşu algoritması yüksek bellek alanına ihtiyaç duymaktadır [98]. Ayrıca k değerine oldukça bağımlıdır. Fakat ilgili k değerine karar verebilmek oldukça zordur. Bu değerin belirlenmesinde deneme yanılma yolu kullanılarak en uygun sonucu veren k değeri seçilir. Tüm bunlara ek olarak öznitelik sayısının çok olduğu yüksek boyutlu veriler üzerinde performansı oldukça düşmektedir [99].

Destek vektör makineleri

Destek vektör makineleri, yüksek bir doğruluk ile çalıştığından ve seçilmiş çekirdek fonksiyonlarına göre oldukça düşük hesaplama maliyeti ile çalışabildiğinden dolayı araştırmacılar tarafından sıklıkla tercih edilir [100-102]. Genellikle oldukça karmaşık ve çok sayıda öznitelik barındıran verilerin analizinde kullanılmaktadır. Destek vektör makineleri n boyutlu bir uzayda bulunan noktaların belirgin bir şekilde ayrılabilmesi için gerekli olan hiperdüzlemi bulmayı amaçlar [103]. Burada bulunan boyut sayısı veride bulunan öznitelik sayısıdır.



Şekil 2.4. Destek vektör makineleri için hiperdüzlemler ve maksimum tolerans

Şekil 2.4 üzerinde görülmektedir ki iki sınıf arasında seçilebilecek birçok muhtemel hiperdüzlem bulunmaktadır. Destek vektör makineleri şekil 2.4 üzerinde de gösterilmiş

olan maksimum toleransı veren hiperdüzlemi bulmayı amaçlar [104]. En yüksek maksimum toleransa sahip olan hiperdüzlem en uygun çözüm olarak kabul edilir. Eğer verinin öznitelik sayısı 2 ise en uygun hiperdüzlem bir çizgi olacaktır. Veri 3 özniteliğe sahip ise hiperdüzlem iki boyutlu bir düzlem olarak tespit edilecektir. 4 ve daha fazla özniteliğe sahip veri için hiperdüzlemi göz önünde canlandırmak oldukça zorlaşmaktadır. Doğrusal destek vektör makineleri doğrusal bir şekilde sınıfların bir birinden ayrıldığı durumlarda kullanılır. Ancak her türlü veri üzerindeki sınıfları doğrusal olarak ayırmak mümkün olmayabilir. Bu durumda doğrusal olmayan destek vektör makineleri kullanılır. Doğrusal olmayan destek vektör makinelerinin hesaplamalar için kullandığı kernel fonksiyonlarına polinom, radyal tabanlı fonksiyon ve PUK fonksiyonu örnek gösterilebilir [105]. Destek vektör makinelerinde kullanılacak kernel fonksiyonunun seçimi ve seçilen kernel fonksiyonunun parametrelerinin belirlenmesinin oldukça zor olmasına ek olarak belirlenen bu değerlere destek vektör makinelerinin performansının doğrudan bağlı olmasından dolayı uygulanmaları oldukça zorlaşabilmektedir.

Navie bayes sınıflandırıcı algoritması

Naive bayes sınıflandırıcı algoritması bayes teoremine dayanan bir olasılıklı sınıflandırma algoritmasıdır [106]. Naive bayes sınıflandırıcı algoritmasında özniteliklerin bir birinden bağımsız olması büyük öneme sahiptir. Çünkü bu algoritmada özniteliklerin hepsinin ilgili kaydın sınıflandırılabilmesi için öneminin aynı olduğu varsayılmaktadır [107]. Tüm bunlara ek olarak öznitelikleri arasında bağımlılık veya benzerlik olmayan bir veri kümesi bulmak oldukça düşük ihtimale sahiptir. Gerçek hayatta üzerinde çalışılan verilerin öznitelikleri arasında genellikle bağımlılık bulunduğundan naive bayes sınıflandırıcı algoritması ile doğru sonuca ulaşmak zorlaşabilmektedir. Fakat ilgili algoritma yinelemeli bir şekilde çalışmadığı için büyük veri kümeleri için sıklıkla tercih edilmektedir.

Kümeleme

Kümele algoritmaları ile verideki her kayıt belirli bir kümeye atılır [88]. Atılan bu kümedeki nesnelerin kendisi ile maksimum benzerlikte olması ve kendi kümesinde olmayan nesneler ile en az benzerlik barındırması beklenir. Bu durum sınıf içi yüksek benzerlik ve sınıflar arası düşük benzerlik ile ifade edilebilir [108]. Benzerlik hesaplaması genellikle uzaklık fonksiyonları kullanılarak yapılır [109]. Kümeleme algoritmasının başarımı kullanılan benzerlik ölçüsüne doğrudan bağlıdır. Ayrıca kümelemede önceden

bilinen bir sınıf etiketi kullanılmadan kayıtlar üzerinde analizler yapılır. Kümelemenin amacı bu sınıf değerlerini oluşturmaktır. Tüm bunlara ek olarak kümele algoritmaları için bölümleyici kümeleme, hiyerarşik kümeleme, yoğunluk tabanlı kümeleme ve ızgara tabanlı kümeleme olmak üzere farklı kümeleme teknikleri bulunmaktadır [110, 111].

Bölümleme tabanlı kümeleme

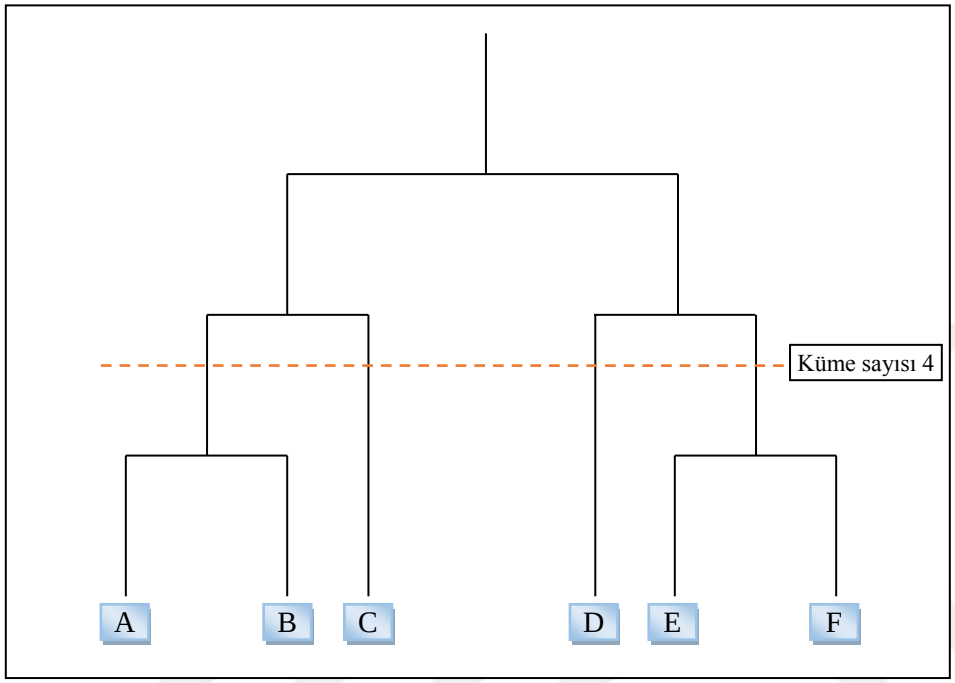
Bölümleme tabanlı kümelemede her kümenin bir nesneye sahip olması gerekmektedir. Buna ek olarak her kayıt yalnızca bir kümeye dahil olabilir. İlgili yöntemde veri belirli adette kümeye bölünür. Bölümleme tabanlı kümelemeye örnek olarak K-means algoritması ve K-medoid algoritması gösterilebilir [108].

Hiyerarşik tabanlı kümeleme

Hiyerarşik kümelemede başlangıç değeri olarak küme sayısına ihtiyaç duyulmamaktadır. Bu kümeleme yöntemi birleştirici hiyerarşik kümeleme yöntemleri ve ayrıştırıcı hiyerarşik kümeleme yöntemleri olmak üzere ikiye ayrılabilir [112, 113]. Birleştirici hiyerarşik kümeleme yöntemleri, yığınsal kümeleme yöntemleri adı ile de literatür taramasında görülmüştür [114, 115]. Birleştirici hiyerarşik kümeleme yöntemlerinde her nokta bir küme olarak kabul edilir ve noktalar kümelerin merkezi olarak belirlenir. Her nokta uzaklık fonksiyonları ile belirlediği en yakın noktayı alarak tek bir küme oluşturur. Bir sonraki adımda uzaklık fonksiyonlarından yararlanarak belirlenen en yakın iki küme tek bir küme olarak kabul edilir. Bu işlem en sonunda bir küme kalana kadar devam eder. Böylece kümeler iteratif bir şekilde birleşerek en büyük kümeyi oluşturur. Oluşturulmuş en küçük kümelerden en büyük kümeye ulaşılana kadar geçen süreçte oluşan kümeler arasındaki hiyerarşi dendogram denilen ağaçlara kaydedilir [116]. Bu ağaçlar üzerinden her iterasyondaki işlemler kolaylıkla görülebilir. Dendogram, oluşturulmuş her küme arasındaki hiyerarşik yapıyı gösterir [108]. Bir ağaç şemasıdır [117]. Şekil 2.5 üzerinde görüldüğü üzere bir hiyerarşik kümeleme sonucunda elde edilmiş dendogram görülmektedir.

İlgili dendogram üzerinde ilk olarak A, B, C, D, E ve F nesnelerinin bir küme olarak seçildiği görülmektedir. Ardından A ve B kümelerinin birleştiği ardından bu birleşmiş küme ile C kümesinin birlikte yeni bir küme oluşturduğu görülmektedir. Benzer şekilde ilk iterasyonda E ve F kümelerinin birleştiği ardından birleşmiş kümenin D kümesi ile birleşerek yeni bir küme oluşturduğu görülmektedir. Son iterasyonda oluşturulmuş bu iki

küme de birleşerek tek bir kümeye ulaşılmıştır. Dendogram üzerinden 3 iterasyon sonunda hiyerarşik veri kümesinin oluşturulduğu görülmektedir. Ayrıca Şekil 2.5 üzerinde de gösterildiği gibi bir çizginin ilgili ağaç üzerine çizilmesi ile istenilen seviyede kaç adet kümenin var olduğu bilgisine ve bu kümelerin neleri barındırdığına ulaşılabilir.



Şekil 2.5. Hiyerarşik kümeleme için örnek dendogram

Ayrıştırıcı hiyerarşik kümeleme yönteminde tek bir kümeden daha öznel kümelere ulaşmak amaçlanır. Birleştirici hiyerarşik kümeleme yönteminde yapılan işlemlerin tam tersi uygulanır. Tüm nesneler tek bir küme olarak kabul edilir. Her kümenin bir elemanı kalana kadar tüm kümeler her iterasyonda uzaklık fonksiyonlarına göre bölünür. Birleştirici hiyerarşik kümeleme yöntemine örnek olarak BRICH ve CHAMELON algoritmaları örnek gösterilirken ayrıştırıcı kümeleme yöntemine DRAGON ve DIANA örnek gösterilebilir [108, 118, 119].

Yoğunluk tabanlı kümeleme

Yoğunluk tabanlı kümelemede kümeler veri içindeki nesnelerin yoğunluklu olduğu bölgeler olarak seçilir [120]. Bu sayede rastgele şekillerdeki kümeler belirlenebilir. Bu yaklaşımda yoğunluklu bölgelerin aksine seyrek bölgelerde bulunan kayıtlar ise gürültü ya da sapan veri olarak kabul edilir [120]. Yoğunluk tabanlı kümeleme algoritmalarına DBSCAN ve OPTICS örnek olarak verilebilir [108].

DBSCAN algoritması için en büyük komşuluk yarıçapına ve ilgili yarıçap içinde bulunması gereken minimum nesne sayısına ihtiyaç duyulmaktadır. Bu parametrelerin farklı değerlerde verilmesi ile aynı veri kümesi içerisinde farklı birçok küme çıkarılabilir. DBSCAN algoritmasının tüm kümelerin yoğunluğunu benzer kabul ettiğinden farklı yoğunluklardaki kümelerin yakalanması zorlaşmaktadır [108]. Kümeler arasındaki yoğunluk farklı arttıkça DBSCAN algoritmasının performansı düşmektedir. Bu dezavantajı gidermek adına en büyük komşuluk yarıçapı değerini kullanmayan OPTICS algoritması geliştirilmiştir [121].

Izgara tabanlı kümeleme

Izgara tabanlı kümeleme ile farklı şekillerdeki kümeler bulunabilir. Oldukça hızlı çalışabilirler [108, 122]. Performansı her düzlemdeki hücre sayısına bağlıdır. Izgara tabanlı kümele algoritmalarına CLIQUE, MAFIA, Wave Cluster ve STING örnek olarak gösterilebilir [108, 123, 124].

Izgara tabanlı kümelemede öncelikle bir dizi ızgara hücresi tanımlanır. Ardından her nesne uygun ızgara hücresine atanır. Tüm nesnelerin atama işlemi bittiğinde her ızgara hücresinin yoğunluğu hesaplanır. Yoğunluk, ilgili hücrede bulunan nesnelerin sayısıdır. Yoğunluk değeri önceden belirlenmiş bir eşik değeri altında kalan hücreler göz ardı edilir. Yoğunluğu eşik değerinden büyük olan hücreler yoğun birim olarak tanımlanır ve yoğun birimlerden bitişik olanlar ile kümeler oluşturulur [125]. Tüm bunlara ek olarak bir yoğun birimin tüm komşuları da yoğun birim ise ilgili hücre çekirdek ızgara, yoğun birimin yalnızca bazı komşuları yoğun birim ise bu durumda ilgili hücre sınır ızgara olarak tanımlanır [125]. Tüm bunlara ek olarak bir yoğun birimin komşularından hiçbiri yoğun birim değilse o zaman bu hücre gürültülü ızgara olarak tanımlanabilir [126].

Birliktelik kuralları

Kural tabanlı bir veri madenciliği metodu olan ve ilişki analizi olarak da bilinen birliktelik kuralları ile verinin sahip olduğu öznitelikler arasındaki gizli kalmış ilişkiler tespit edilir [127]. Veri madenciliğinde birliktelik kuralları mühendislik, eğitim ve tıp gibi birçok alanda sıklıkla kullanılmaktadır [48]. Apriori algoritması, Eclat algoritması, Tertius algoritması ve FP-Growth algoritması çoğunlukla tercih edilen birliktelik kuralı algoritmalarından bazılarıdır [128]. İlgili algoritmaların analiz sonuçlarının doğruluğu ile kullanılan eğitim setinin büyüklüğü arasında doğru orantı olduğu varsayılabilir. Dolayısı

ile birliktelik kuralları algoritmaları ne kadar çok kayıt üzerinde çalışırsa üretilen kuralların o kadar doğru olduğu görülebilir. Ancak çok büyük veri kümesi kullanılması ile öğrenme süresi büyüyeceğinden veri kümesini en iyi ifade edebilecek örneklem üzerinde çalışılabilir.

Bir birliktelik kuralı "eğer_öznitelik_dizisi → öyleyse_öznitelik_dizisi" olarak ifade edilir. eğer_öznitelik_dizisi ve öyleyse_öznitelik_dizisi değerleri bir öznitelikten oluşabileceği gibi birden çok öznitelikten de oluşabilir. Örneğin bir market verisinden çıkarılmış olduğu varsayabileceğimiz "{domates, biber, soğan} → {salça}" birliktelik kuralında eğer_öznitelik_dizisi değeri birden çok öznitelikten oluşurken öyleyse_öznitelik_dizisi değeri yalnızca bir öznitelik barındırmaktadır. İlgili birliktelik kuralı incelendiğinde verisi incelenen markette domates, biber ve soğan ürünlerini satın alacak olan müşteriler oldukça yüksek bir olasılık ile salçayı da satın alacağı bilgisine ulaşılabilir. Bu cümleden yola çıkılarak birliktelik kurallarının sol tarafının "eğer", sağ tarafının ise "öyleyse" olduğu düşünülebilir. Dolayısı ile bu birliktelik kuralı şu şekilde de okunabilir: Eğer bir müşteri domates, biber ve soğan satın alırsa büyük olasılıkla salça da satın alacaktır.

Ortaya çıkarılmış birliktelik kurallarının olasılığı destek ve güven değerleri kullanılarak hesaplanır [129]. Destek sayısı incelenen kayıtlardan toplam kaç tanesinde kuralı çıkarılmış özniteliklerin birlikte görüldüğünün sayısıdır. Destek değeri ise destek sayısının tüm kayıtlara olan oranıdır. Örneğin her müşterinin yalnızca bir defa alışveriş yaptığı varsayılan bir market verisinden üretilen "{domates, biber, soğan} → {salça}" birliktelik kuralı ele alındığında destek değeri domates, biber, soğan ve salçayı birlikte satın alan müşteri sayısıdır. Destek değeri ise domates, biber, soğan ve salçayı birlikte satın alan müşteri sayısının tüm müşteri sayısına oranıdır. Güven değeri ise eğer_ öznitelik_dizisi değerinde bulunan özniteliklerin birlikte görüldüğünde öyleyse_öznitelik_dizisi değerinde bulunan özniteliklerin de birlikte görülme sayısının tüm eğer_ öznitelik_dizisi değerinde bulunan özniteliklerin birlikte görülme sayısına oranı ile ifade edilir. Aynı örnek için güven değeri domates, biber, soğan ve salça alan müşteri sayısının tüm domates, biber ve soğan alan müşterilere olan oranıdır. Yani domates, biber ve soğan alan müşterilerin yüzde kaçının domates, biber, soğan ve salça aldığının hesaplanması ile elde edilir.

Bulunan bir kuralın kabul edilebilir ve yaygın bir birliktelik kuralı olarak seçilebilmesi için ilgili kuralın destek ve güven değerlerinin minimum destek eşik değerinden ve minimum güven eşik değerinden büyük olması veya eşit olması gerekmektedir [130]. İlgili eşik

değerlerini sağlayan kurallar “eğer_öznitelik_dizisi → öyleyse_öznitelik_dizisi [destek = %..., güven = %...]” şeklinde ifade edilebilir. Daha öncede örnek olarak verilen “{domates, biber, soğan} → {salça}” kuralının destek değerinin 0,10 ve güven değerinin 0,9 olduğu varsayılırsa destek ve güven değerleri ile birlikte “{domates, biber, soğan} → {salça} [destek = %10, güven = %90]” şeklinde yazılabilir. Bu durumda ilgili kural “eğer bir müşteri domates, biber ve soğan satın alırsa %90 olasılıkla salça da satın alacaktır” şeklinde okunabilir.

2.1.4. Veri tabanından bilgi keşfi uygulama alanları

Günümüzde veri madenciliği, dijital verilerin bulunduğu hemen her yerde kullanılabilmesinden dolayı oldukça geniş bir uygulama alanına sahiptir [131]. Veri madenciliğinin göze çarpan örnekleri pazarlama, yatırım, sahtekârlık tespiti, üretim sanayi ve veri temizleme alanlarında yoğunlaşmaktadır [32]. Bunlara ek olarak veri madenciliği bilgisayar oyunlarında, insan haklarının korunmasında, siber güvenlikte, eğitimde, iş dünyasında, mühendislik ve bilim alanlarında da sıklıkla tercih edilmektedir. [48, 131-133]

Walmart şirketi dünya çapında mağazaları bulunan bir Amerikan market zinciridir [134]. Günde 20 milyondan fazla satış işlemi kayıt altına alınmakta ve merkezi bir veri tabanında toplanmaktadır [135]. Bu gibi veriler iş dünyasında müşterilerin satın alma eğilimlerini açığa çıkarmak, ilgi çekici kapmanyalar oluşturabilmek, müşterileri benzer davranışlara göre kümelemek gibi daha birçok amaç için kullanılabilmektedir. Walmart verileri üzerinde gerçekleştirilen bir veri tabanından bilgi keşfi uygulaması ile veri madenciliği teknikleri kullanılarak müşterilerin satın alma eğilimlerine ait örüntülerin yakalanması amaçlanmıştır [136]. Ortaya çıkan birliktelik kurallarından bazıları tahmin edilebilir kurallar olmasına karşın bazı kurallar ise tamamen sürpriz olmuştur. Tahmin edilebilir kurallara “cin satın alan bir müşteri yüksek olasılık ile tonik de satın alacaktır.” gibi kurallar örnek olarak gösterilebilir. Sürpriz olarak nitelendirilen ve ilgili veriden çıkarılmış en çarpıcı kural ise “eğer bir müşteri bebek bezi satın alırsa yüksek olasılıkla bira da satın alacaktır” birliktelik kuralıdır. Bu çalışma ile veri madenciliğinin asıl amacının bilinen örüntüleri değil veri içinde gizli kalmış örüntüleri bulma amacı taşıdığı görülmüştür. Veri madenciliği algoritmaları ile veri içinde gizli kalmış örüntülerin bulunabileceği bir kez daha açık bir şekilde görülmüştür.

Overwatch mayıs 2016 da piyasaya sürülmesinden itibaren hızlı bir şekilde popülerlik kazanmış çevrim içi bir oyundur [137, 138]. Mevcut popülaritesine rağmen yeni başlayan oyuncuların oyundaki kahramanları nasıl kullanacağı konusunda problem yaşadığı görülmesi üzerine geliştirilen veri madenciliği algoritması ile Overwatch oyun verileri analiz edilmiştir [138]. İlgili çalışma ile mevcut oyuncular hakkında bilgi edinme ve oynayış örüntülerinin keşfedilmesi amaçlanmıştır. Kazanan oyuncuların stratejileri analiz edilmiş ve bu oyuncu hareketleri ile ilişkili örüntüler çıkarılır. Çıkarılan bu örüntüler kullanılarak yeni başlayan oyunculara oyundaki kahramanları nasıl kullanabileceği hakkında önerilerde bulunulur. İlgili çalışma ile veri madenciliği kullanılarak çevrim içi oyun verilerinin analiz edilebileceği ve bu analizler sonucunda kullanıcılara oynayış teknikleri hakkında önerilerde bulunulabileceği gösterilmiştir.

Üniversite öğrencilerinin akademik olarak başarılı olabilmeleri için bir veri madenciliği uygulaması gerçekleştirilmiştir [139]. İlgili çalışmada üniversite öğrencilerinin büyük çoğunluğunun ilk yılında başarısız olduğu için bu durumun engellenmesi amacıyla veri madenciliği tekniklerinden faydalanılarak öğrencileri oldukça erken bir aşamada üç grup üzerinden sınıflandırmak amaçlanmıştır. Bu gruplar düşük, orta ve yüksek riskli olarak tanımlanmıştır. Yüksek riskli grup, üniversitenin ilk yılında akademik olarak yüksek olasılıkla başarısız olacak öğrencileri temsil etmektedir. Düşük riskli grup ise üniversitenin ilk yılında yüksek olasılıkla başarılı olabilecek öğrencileri temsil etmektedir. Çalışma, 2003-2004 öğretim yılı kasım ayında öğrencilere sorulmuş 533 soru üzerinden gerçekleştirilmiştir. Veri madenciliği tekniklerinden karar ağaçları, rastgele ormanlar, sinir ağları ve doğrusal diskriminant analizi kullanılarak sonuçlar karşılaştırılmıştır. Sonuçlar incelendiğinde yüksek riskli öğrencileri bulmak için karar ağaçlarının, orta riskli öğrencilerin belirlenmesi için rastgele ormanların, düşük riskli öğrencilerin tespit edilmesi için ise doğrusal diskriminant analizinin kullanılabileceği görülmüştür.

2.2. Saldırı Tespit Sistemi

Saldırı tespit sistemleri (STS), zararlı aktiviteleri tespit etmek amacıyla canlı veya kaydedilmiş ağ trafiği üzerinde analizler yapan güvenlik sistemlerdir [140]. Analizler hem gelen trafik hem de giden trafik üzerinde gerçekleştirilir. Bu analizler neticesinde bilgiyi ve bilgi sistemlerini tehlikeye düşürebilecek her türlü kötü niyetli aktivite otomatik olarak tespit edilerek alarm üretilmekte ve/veya rapor oluşturulmaktadır [10, 141].

STS, tehditleri engellemeye odaklanmak yerine yalnızca tespit etmeyi amaçlar [26]. Bu nedenle canlı trafik üzerinde kullanıldığı durumlarda genellikle trafiğin bir kopyası üzerinde çalışmaktadır [142]. Gönderen ve alıcı arasına yerleşerek trafik üzerinde yönetimsel işlemler yapmasına ihtiyaç duyulmadığından kopya trafiğin işlenmesi STS'nin işlevleri için problem oluşturmamaktadır. Canlı trafiğin kopyası üzerinde analizlerin yapılmasıyla STS'nin canlı trafiğin hızı üzerinde olumsuz etkide bulunmaması sağlanmaktadır.

2.2.1. Saldırı tespit sistemi sınıflandırmaları

Saldırı tespit sistemleri bulundukları konuma göre sınıflandırılmaktadırlar. Bu kapsamda saldırı tespit sistemleri anasistem tabanlı ve ağ tabanlı saldırı tespit sistemleri olarak iki sınıfa ayırmak mümkündür [143].

Ağ tabanlı saldırı tespit sistemleri

Ağ tabanlı saldırı tespit sistemleri ile ağ üzerinden akan trafik üzerinde incelemeler yapılabilir [11]. İncelemeler hem gelen trafik hem giden trafik üzerinde gerçekleştirilir. Her paket analiz edilerek zararlı aktiviteler tespit edilmeye çalışılır. Bu tip saldırı tespit sistemleri genellikle bir güvenlik duvarının arkasına yerleştirilir. Böylece güvenlik duvarı tarafından uygulanmış filtrelerden kurtularak gözden kaçan zararlı aktiviteler yakalanır. Ağ tabanlı saldırı tespit sistemleri çevrim içi saldırı tespit sistemleri ve çevrimdışı saldırı tespit sistemleri olmak üzere ikiye ayrılabilir [140]. Gerçek zamanlı saldırı tespit sistemleri olarak da bilinen çevrim içi saldırı tespit sistemleri ile canlı trafik üzerinde inceleme yapabilen saldırı tespit sistemleri ifade edilir. Çevrim dışı saldırı tespit sitemleri ile ise kaydedilmiş trafik üzerinde incelemeler yapan saldırı tespit sistemleri temsil edilir.

Ağ tabanlı saldırı tespit sistemleri düşük maliyet, kolay kurulum, ağ tabanlı saldırıların tespiti, saldırı kayıtlarının gerçek zamanlı olarak işlenmesi, gerçek zamanlı tespit ve gerçekleştirilemeyen saldırıların dahi tespit edilmesi gibi birçok avantaja sahiptir [143]. İçinde bulundukları ağın tamamı üzerinde analizler gerçekleştirdiklerinden ve dolayısı ile tüm sistemi kontrol edebildiklerinden maliyet oldukça düşmektedir. Ayrıca bir ağ tabanlı saldırı tespit sistemini mevcut ağa dahil etmek için ise alt yapı üzerinde bir değişikliğe ihtiyaç duyulmamaktadır. Bu durum kurulumlarını oldukça kolaylaştırmaktadır. Tüm bunlara ek olarak saldırganlar bir sistemi ele geçirdiklerinde veya ilgili saldırıyı tamamladıklarında arkalarında iz bırakmamak için kendileri ile alakalı tüm kayıtları silmek

isteyebilirler. Ağ tabanlı saldırı tespit sistemlerinin trafiği sürekli izleyerek gerçek zamanlı olarak işlem yapmasından dolayı saldırganlar birçok kanıtı geride bırakmak zorunda kalırlar. Buna ek olarak birçok anasistem tabanlı saldırı tespit sistemi işletim sistemi günlük dosyaları üzerinde kontroller gerçekleştirmekte fakat saldırganlar genellikle bu günlük dosyaları üzerinde de değişiklik yaptığından böyle durumlarda saldırılar tespit edilemeyebilmektedir. Ağ tabanlı saldırı tespit sistemleri saldırıları gerçek zamanlı olarak tespit edilebilmesinden dolayı ilgili saldırılar gerçekleşir gerçekleşmez yakalanabilmektedir. Bu durum gereken önlemleri almak için harekete geçme zamanını sağlar. Böylece birçok saldırı henüz başarıya ulaşmadan engellenebilir. Ağ tabanlı saldırı tespit sistemlerinin genellikle güvenlik duvarı arkasında konumlandırılmasına rağmen güvenlik duvarının önüne de kurulumu yapılabilir. Böyle durumlarda güvenlik duvarı tarafından engellenen saldırılar hakkında da bilgi toplanabilir.

Anasistem tabanlı saldırı tespit sistemleri

Anasistem tabanlı saldırı tespit sistemleri her cihaz üzerinde ayrı çalışarak yalnızca ilgili cihaza gelen ve giden trafik üzerinde incelemelerde bulunulur [11]. Anasistem tabanlı saldırı tespit sistemleri yalnızca trafiği izlemek yerine işletim sistemi üzerinde de analizlerde bulunabilirler [143]. Büyük ağlarda içerden kaynaklı olan saldırıların yakalanmasında oldukça etkilidir [26].

Anasistem tabanlı saldırı tespit sistemleri saldırının başarıya ulaşıp ulaşmadığının tespiti, sistem aktivitelerinin takip edilebilmesi ve ayrı bir donanım gerektirmemesi gibi birçok avantaja sahiptir. Anasistem tabanlı saldırı tespit sistemleri işletim sistemi günlükleri üzerinde de inceleme yapabilme yeteneğine sahip olduğundan maruz kalınmış saldırıları tespit edebilir ve günlük dosyaları gibi kaynaklardan ilgili saldırının başarıya ulaşıp ulaşmadığını doğrulayabilir [11]. Ayrıca anasistem tabanlı saldırı tespit sistemleri dosya erişim aktiviteleri, yeni uygulamaların yüklenmesi, oturum açma veya kapama gibi aktiviteleri de inceleyerek analizler gerçekleştirebilir [143]. Tüm bunlara ek olarak anasistem tabanlı saldırı tespit sistemleri bir yazılım olarak çalıştığından ayrıca bir donanıma ihtiyaç duymamaktadır. Bu durum maliyetlerin düşürülmesi açısından önem arz etmektedir.

2.2.2. Saldırı tespit sisteminde kullanılan yöntemler

Saldırı tespit sistemleri, zararlı aktivitelerin tespiti amacıyla kullandığı tekniklere göre imza tabanlı ve anomali tabanlı olarak iki ana gruba ayrılır [12, 144]. İmza tabanlı saldırı tespit sistemleri aynı zamanda bilgiye dayalı saldırı tespit sistemleri olarak, anomali tabanlı saldırı tespit sistemleri ise davranış tabanlı saldırı tespit sistemleri olarak da bilinmektedir [14, 145].

İmza tabanlı saldırı tespit sistemleri

İmza tabanlı saldırı tespit sistemleri zararlı aktiviteleri tespit edebilmek için imza veri tabanına ihtiyaç duyar [13]. İmza veri tabanı bilinen saldırıların tanınabilmesi için gerekli olan bilgileri içerir. Bir aktiviteyi analiz edecek olan STS öncelikle ilgili aktiviteyi imza veri tabanında bulunan imzalar ile karşılaştırır. Analiz edilen aktivite, herhangi bir imza ile eşlenirse saldırı olarak tanımlanır. Zararlı aktivitelerin tespitindeki başarı tamamıyla imza veri tabanına bağlıdır. Dolayısı ile imza veri tabanının mümkün olduğunca mevcut tüm saldırılar hakkında bilgi içermesi ve bu bilgilerinde güncel olarak tutulması gerekmektedir.

İmza tabanlı saldırı tespit sistemlerinde kullanılan imzaların ilgili saldırılar için önceden hazırlanması gerektiğinden yeni çıkan saldırı tipleri için tespit yapılamamaktadır. Dolayısı ile bilinmeyen saldırı çeşitleri için herhangi bir koruma sağlanmamaktadır [146]. Ayrıca imzaların oluşturulması oldukça zor bir şekilde gerçekleştirilebilir.

Anomali tabanlı saldırı tespit sistemleri

Anomali tabanlı saldırı tespit sistemleri mevcut ağın olağan durumunu öğrenir [25]. Ardından sistem üzerinde gerçekleşecek olağan dışı her aktivite saldırı şüphesi ile raporlanır [17]. İlgili aktiviteleri normal veya anormal olarak tanımlar [147]. Mevut ağın olağan durumunun öğrenilmesinde genellikle ağda bulunan makinelerin veya tüm ağın trafik yükü, kullanılan protokoller veya tipik paket büyüklükleri gibi veriler kullanılır. Bu veriler için beklenen durumlar tanımlanır. Bu durumlarda gözlenen bariz değişiklikler anomali tabanlı saldırı tespit sistemi tarafından zararlı aktivite olarak tanımlar.

Saldırıların tespiti için bir imza veri tabanı kullanmadığından dolayı yalnızca bilinen saldırıların tespitinde değil aynı zamanda bilinmeyen saldırılarında tespitinde oldukça başarılır. Ancak çok fazla yanlış alarm üretmeye müsaittir [15].

Klasik anomali tespiti yerine veri madenciliği tekniklerinden faydalanılarak yanlış alarm sayısı düşürülmekte ve daha doğru sonuçlar veren sistemler oluşturulabilmektedir. [18, 23, 25]

2.2.3. Saldırı tespit sistemleri sınırlılıkları

Saldırı tespit sistemleri şifrelenmiş paketlerin içeriğini çözemeyeceğinden bu paketlerin taşıdığı verileri analiz edemez [148]. Verinin güvenli bir şekilde taşınması için kullanılan şifreli paketlerin gün geçtikçe daha yaygın duruma gelmesi saldırı tespit sistemleri için büyük bir dezavantaj oluşturmaktadır.

Ağ paketleri üzerinde yapılan incelemelerde paket içindeki bilgiler kullanılarak analizler gerçekleştirilir. Ancak ağ paketleri değiştirilebilirdir ya da kasıtlı olarak farklı verilerle oluşturulabilir [149, 150]. Bu durum aktivitenin zararlı ya da zararsız olduğunun tespitini zorlaştırmaktadır. Örneğin bir trafik paketinin IP adresi bilgisi kullanılarak yapılan analizlerde, kasıtlı olarak paketlerin IP adresi bilgisi değiştirilerek yanlış alarm oluşturulabilir ya da zararlı aktivitelerin gözden kaçırılması sağlanabilir.

İmza tabanlı saldırı tespit sistemleri, imza veri tabanının sürekli olarak güncellenmesine ihtiyaç duyar. Buna ek olarak yeni bir saldırı türü tespit edildiğinde, imza veri tabanı güncellenene kadar ilgili saldırıya sistemler açık bir şekilde kalır. İmzaların üretilmesin de yaşanan zorluklardan dolayı yeni saldırıya karşı alınabilecek önlemlerin uygulanması gecikebilir. Ayrıca kullanılan imzaların başarısı doğru bir şekilde test edilmelidir. Kullanılacak yanlış imzalar oldukça fazla yanlış alarma sebep olacaktır.

Trafikte bulunan gürültülü veriler, saldırı tespit sistemleri tarafından yanlış alarm üretilmesine sebep vermektedir [151]. Genellikle bu durum üretilmiş alarmlar üzerinde ikincil bir analize gereksinim oluşturur. Dolayısı ile yanlış alarmlar yüzünden kullanıcılar üzerinde fazladan iş yükü oluşturabilmektedir.

2.3. Bal Küpü Sistemleri

Ayılar için kurulan tuzakların cazip hale getirilmesi için kullanılan bal küplerinden adını almış olan bu sistemler, saldırganların dikkatini çekerek saldırıları kendilerinde yönlendiren ve tuttuğu günlükler ile saldırganlar ve teknikleri hakkında bilgiler toplayan sistemlerdir [26]. Genellikle herhangi bir trafik üretmemelerine ek olarak herhangi bir trafiğinde de yönlendirilmediği bu sistemler, değişen saldırgan tekniklerine ve yazılım

teknolojilerine adapte olabilmek için çeşitli trafik üretebildiği veya yönlendirildiği durumlar ile karşılaşılmaktadır [26, 152].

2.3.1. Bal küpü sistemi sınıflandırmaları

Bal küpü sistemleri düşük etkileşimli ve yüksek etkileşimli olmak üzere iki ana grup altında incelenebilir [27]. Düşük etkileşimli bal küpü sistemleri servisleri simüle eder [26]. İçlerinde gerçek servisler barındırmazlar. Genellikle yazılım olarak kurulurlar. Bu yüzden kurulumları yüksek etkileşimli bal küpü sistemlerine nazaran kolaydır. Servislerin gerçekten çalıştırmadıklarından dolayı sistem kaynaklarını oldukça düşük olarak kullanırlar. Tüm bu nedenlerden dolayı tek bir sistem üzerinde birden fazla düşük etkileşimli bal küpü çalıştırılabilir. Tüm bunlara ek olarak düşük etkileşimli bal küpü sistemleri gerçek servislerin sahip olduğu tüm komutları desteklemeyebilir. Bu durum saldırganın düşük etkileşimli bal küpü sisteminin desteklemediği bir komutu çalıştırması ile üzerinde koşulan yazılımın bir bal küpü sisteminde olduğu anlaşılabilir. Destekledikleri komutların sınırlı olmasından ve servisleri yalnızca simüle ettiklerinden dolayı saldırganların teknikleri hakkına toplanacak bilgiler oldukça sınırlıdır. Tüm bunlara ek olarak bilinen saldırıları tekniklerini kullanan saldırganları tespit etmek için sıklıkla tercih edilirler.

Yüksek etkileşimli bal küpü sistemleri karmaşık yapıya sahiptirler. Kurulumları düşük etkileşimli bal küpü sistemlerine nazaran oldukça zordur. Gerçek servis ve sistemleri barındırırlar [26]. Dolayısı ile saldırganların teknikleri hakkında oldukça detaylı bilgi elde edilebilir. Bilinen saldırıların tespiti yerine sıklıkla sıfırıncı gün saldırılarının tespiti ve saldırganın yöntemleri hakkında bilgi toplamak amacıyla kullanılabilir.

Düşük etkileşimli bal küpü sistemlerine Kojoney, Honeyd, DemonHunter, KFSensor, Specter, HoneyPy, Dionaea ve Honeystick örnek olarak verilebilirken yüksek etkileşimli bal küpü sistemlerine Capture HPC, HoneyBrid, Sebek ve Symantec Decoy Server örnek olarak gösterilebilir.

2.3.2. Bal küpü sistemlerinin önemi

Bal küpü sistemlerinde temel varsayım hiçbir sistemin bal küpleri ile herhangi bir iletişime ihtiyaç duymayacağıdır. Ağ üzerinde bulunan diğer sistemlerin dahi bal küpü sistemi içeren makinenin varlığından haberdar olunması beklenmez. Dolayısı ile bal küpü

sistemini tespit etmiş ve iletişimde olduğu görülen tüm sistemlerin zararlı bir aktivitede bulunduğu kabul edilebilir. Bal küpü günlükleri incelenerek saldırganlar tespit edilebilir ve saldırı teknikleri analiz edilebilir. Bazı bal küpü sistemleri saldırıları başarılı olmuş gibi simüle ederek saldırganın bir açıklıktan faydalandıktan sonra ele geçirdiği sistemde neler yaptığı gibi bilgiler elde edilebilir [26].

Bal küpü sistemleri yüksek işlemci gücüne veya yüksek bellek alanına ihtiyaç duymadığından düşük sistem kaynakları ile çalışabilirler [153]. Bu sayede bir sunucu üzerine birden çok bal küpü sistemi de çalıştırılabilir. Ayrıca temel olarak bal küpü sistemleri ile iletişime geçen her türlü sistemin zararlı aktivite olduğu varsayıldığından yanlış alarm üretmeleri teknik olarak imkânsızdır. Hali hazırda bal küpü sisteminin orada bulunduğunun dahi tespit edilmesi beklenemezken iletişime geçen aktörün kötü amaçlı kullanıcı olduğu açıktır.

2.4. Siber Tehdit İstihbaratı

Siber tehdit, bir veya daha fazla açıklığı sömürerek güvenlik önlemlerini aşabilecek her türlü tehlike olarak ifade edilebilir [154]. Siber tehdit istihbaratı ise siber tehditlerin tanımlanabilmesi ve yakalanabilmesi için gerekli olan bilgi olarak veya siber güvenliğin sağlanabilmesi için kullanılabilecek her türlü veri olarak ifade edilebilir [1, 2]. Bir STİ, komuta kontrol sunucusunun IP adresi, zararlı bir dosyasının özet bilgisi, oltalama saldırısı için kullanılan bir mail ya da bir zararlı aktivitede bulunan saldırgan ve saldırı tekniği hakkında bilgi olabilir. [155]

2.4.1. Siber tehdit istihbaratı sınıflandırmaları

Faaliyetlerine 1 Ekim 2016 tarihinde başlayan Birleşik Krallık Ulusal Siber Güvenlik Merkezi, STİ'leri taktiksel, teknik, operasyonel ve stratejik olmak üzere 4 gruba ayırmaktadır [5]. Bu sınıflandırma ile STİ'ler barındırdıkları bilgilere ve ilgili bilgilerin hedef kitlelerine göre gruplara ayrılmaktadır. MWR InfoSecurity tarafından yapılmış çalışma [5] ışığında taktiksel, teknik, operasyonel ve stratejik siber tehdit istihbaratı grupları açıklanmıştır.

Taktiksel STİ saldırganların kullandığı araçlar ve yöntemleri hakkında bilgiler içerir. Tehdit aktörlerinin saldırıyı nasıl gerçekleştirdiği hakkındaki bilgi paylaşımlarını amaçlar. Bu tip STİ ilgili saldırı yöntemini engellemek için kullanılabilecek önlemleri de içerebilir.

Genellikle bu istihbarat bilgisi uzun ömürlüdür. Saldırganların Mimikatz gibi uygulamalar vasıtası ile kullanıcıların açık halde bulunan giriş bilgilerini ele geçirerek PsExec gibi araçlar vasıtası ile bu bilgileri kullandığı bilgisi taktiksel STİ için örnek olarak gösterilebilir.

Teknik STİ saldırganları ya da zararlı yazılımları tespit etmek için ihtiyaç duyulan bilgileri içerir. Bu tip STİ bir komuta kontrol sunucunun IP adresi bilgisi veya zararlı bir dosyanın özet bilgisi örnek olarak gösterilebilir. Ancak genellikle kısa ömürlüdürler. Çünkü IP adresi bilgisi ya da dosya özet bilgisi saldırganlar tarafından kolaylıkla değiştirilebilirdir.

Stratejik STİ teknik olmayan bilgiler içerdiğinden daha çok yönetim seviyesindeki personel tarafından kullanılmaktadır. Siber saldırıların finansal etkileri ve saldırı eğilimleri hakkında bilgiler içerir.

Operasyonel STİ, kurum ve kuruluşlara karşı yapılacak olan saldırılar hakkında bilgiler içerir. Bu tür STİ oldukça nadir olarak paylaşılabilmektedir. Genellikle hedef ilgili kurum ve kuruluşların üst düzey güvenlik yöneticileridir.

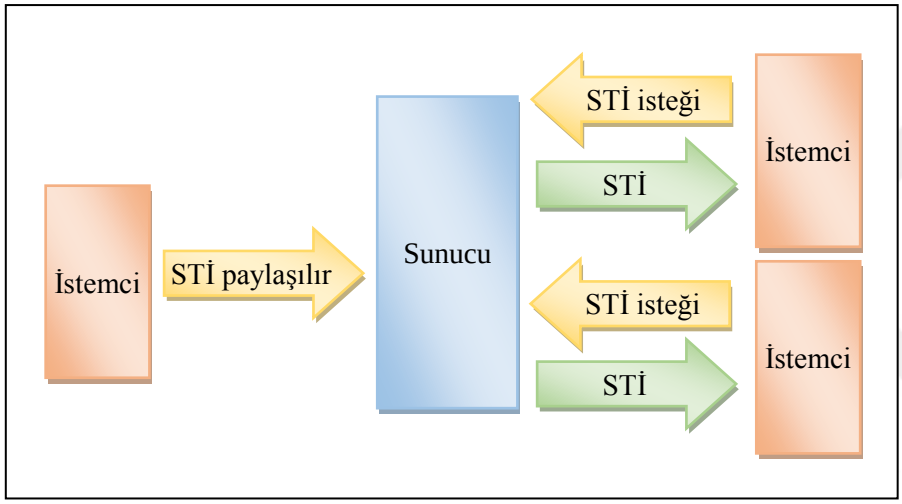
2.4.2. Siber tehdit istihbaratı standartları

STİ'leri ifade etmek için birçok standart kullanılmaktadır. Bu standartların kullanılması ile tüm paydaşlar arasında aynı dilin konuşulması sağlanır. Böylece oluşturulmuş STİ herkes tarafından anlaşılabilir ve kullanılabilir olması mümkün kılınır. Bu standartlara TAXII, Vocabulary for Event Recording and Incident Sharing (VERIS), CYBOX ve STIX örnek olarak gösterilebilir [6].

Açık kaynak kodlu olan TAXII, STİ'lerin güvenli bir şekilde taşınması için kullanılan ve klasik istemci – sunucu mimarisi ile çalışan bir uygulama katmanı protokolüdür [156, 157, 158]. Şekil 2.6 üzerinde çalışma prensibi gösterilmiştir. RESTful servisler ile çalışmaktadır [156]. STİ üreten bir kullanıcı, ilgili verileri sunucuya gönderebilmekte ve bu sunucu üzerinden ise diğer istemciler ilgili STİ'e ulaşabilmektedir.

Bir taşıma protokolü olan TAXII'nin aksine STIX, CYBOX ve VERIS STİ tanımlamak için kullanılır. Diğer bir ifadeyle kazanılmış bilgiler bu standartlar ile STİ olarak ifade edilir.

STIX ilk sürümünde STİ'leri XML kullanarak oluşturmaktaydı [159]. Resim 2.1 üzerinde XML veri formatı kullanan STIX 1.0 versiyonunda üretilmiş STİ gösterilmiştir. Ancak STIX 2.0 versiyonunda XML yerinde JSON veri formatı kullanılmıştır [159]. JSON veri formatının seçilmesinde XML veri formatına nazaran kullanımının kolay olması ve geliştiriciler tarafından daha sık tercih edilmesi gibi nedenler etkili olmuştur. Ayrıca CYBOX standardı da ilk versiyonunda XML veri formatı kullanmaktaydı. Ancak STIX 1.0 standardı ve CYBOX standardı STIX 2.0 versiyonunda birleştirilerek tek bir standart haline getirilmiştir [159].



Şekil 2.6. TAXII protokolü istemci – sunucu mimarisi

```
<stix:Indicator id="example:indicator-01"
               timestamp="2017-02-09T12:11:11.415000+00:00"
               xsi:type='indicator:IndicatorType'>
 <indicator:Title>HTRAN Hop Point Accessor</indicator:Title>
</stix:Indicator>
<stix:TTPs>
 <stix:Kill_Chains>
  <stixCommon:Kill_Chain id="stix:TTP-02"
                         name="mandiant-attack-lifecycle-model">
  <stixCommon:Kill_Chain_Phase name="establish-foothold"
                               phase_id="stix:TTP-03"/>
 </stix:Kill_Chains>
</stix:TTPs>
<indicator:Observable id="example:Observable-04">
 <cybox:Object id="example:Object-05">
  <cybox:Properties xsi:type="AddressObj:AddressObjectType"
                    category="ipv4-addr">
  <AddressObj:Address_Value condition="Equals">10.1.0.0/15
  </AddressObj:Address_Value>
 </cybox:Object>
</indicator:Observable>
```

Resim 2.1. STIX 1.0 versiyonu ile üretilmiş STİ [159]

CYBOX standardının ilk versiyonu kullanılarak oluşturulmuş örnek bir STİ ise Resim 2.2 üzerinde gösterilmiştir. Resim 2.3 üzerinde ise JSON veri formatı kullanan STIX 2.0 versiyonu ile üretilmiş STİ gösterilmiştir.

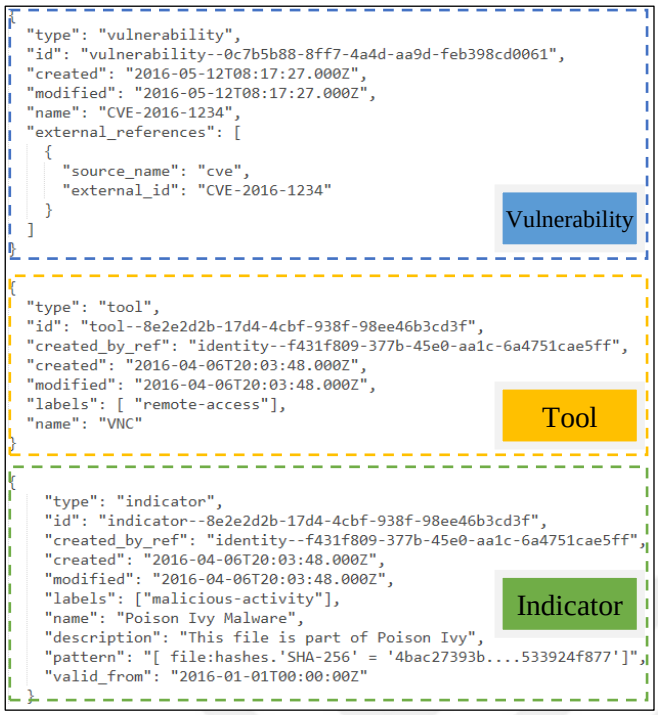
```
<cybox:Observables
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:cybox="http://cybox.mitre.org/cybox-2"
    xmlns:IPAddressObj="http://example.com/objects#IPAddressObject-1"
    xmlns:example="http://example.com/"
    xsi:schemaLocation="
    http://cybox.mitre.org/cybox-2 http://cybox.mitre.org/XMLSchema/core/2.1/cybox_core.xsd
    http://example.com/objects#IPAddressObject-1 http://example.com/some/path/to/IPAddressObject.xsd"

    cybox_major_version="2" cybox_minor_version="1" cybox_update_version="0">
    <cybox:Observable id="example:Observable-6dde4f72-cca0-44f0-8206-f381accf6b89">
        <cybox:Object>
            <cybox:Description>An IP Address</cybox:Description>
            <cybox:Properties xsi:type="IPAddressObj:IPAddressObjectType" category="ipv4" is_source="true">
                <IPAddressObj:IP_Address_Value>10.0.0.1</IPAddressObj:IP_Address_Value>
            </cybox:Properties>
        </cybox:Object>
    </cybox:Observable>
</cybox:Observables>
```

Resim 2.2. CyBOX ile üretilmiş STİ [160]

STIX 2.0 formatında Resim 2.3 üzerinde görüldüğü üzere nesnelerin tipleri bulunmaktadır. Bu standartta saldırı örüntüleri, kimlikler, göstergeler, zararlı yazılımlar, raporlar, tehdit aktörleri, araçlar ve açıklıklar gibi birçok nesne tipi barındırılmaktadır [159]. Kötü amaçlı siber tehditlerin tespit edilebilmesi için kullanılacak istihbarat bilgileri Resim 2.3 üzerinde olduğu gibi göstergeler ile ifade edilir.

```
{
 "type": "indicator",
 "id": "indicator--01",
 "created": "2017-02-09T12:11:11.415000Z",
 "modified": "2017-02-09T12:11:11.415000Z",
 "name": "HTRAN Hop Point Accessor",
 "pattern": "[ipv4-addr:value =
                    '10.1.0.0/15']",
 "labels": [ "malicious-activity" ],
 "valid_from": "2015-05-15T09:00:00.000000Z",
 "kill_chain_phases": [
   {
     "kill_chain_name":
       "mandiant-attack-lifecycle-model",
     "phase_name": "establish-foothold"
   }
 ]
}
```

Resim 2.3. STIX 2.0 versiyonu ile üretilmiş STİ [159]



Resim 2.4. Tool, Vulnerable ve Indicator STIX bileşenleri için örnek kayıtlar [159]

STIX oldukça zengin bir dil olmasına karşın bünyesinde tanımlanmış birçok bileşen bulunmaktadır [161]. Bunlardan bazıları Çizelge 2.1 üzerinde gösterilmiştir. Resim 2.4 üzerinde ise gösterge (indicator), araç (tool) ve açıklık (vulnerability) bileşenlerinin örnekleri gösterilmiştir.

Çizelge 2.1. STIX bileşenleri [159]

Bileşen	**Amaç**
Saldırı Düzeni (Attack Pattern)	Saldırıları kategorilere ayırmak, uygun görüldükleri kalıplar için saldırıları genellemek ve saldırıların nasıl gerçekleştirildiği hakkında ayrıntılı bilgi sağlamak için kullanılır
Savaş (Campaign)	Belirli bir hedef kümesine karşı belirli bir süre boyunca yapılan kötü amaçlı etkinlikleri tanımlayan bir dizi olumsuz davranışı ifade etmek için kullanılır
Gösterge (Indicator)	Şüpheli veya kötü amaçlı siber etkinlikleri tespit etmek için kullanılabilecek düzenli ifadelerin paylaşılması için kullanılır
Saldırı Kümesi (Intrusion Set)	Tek bir kuruluş tarafından yönetildiğine inanılan ortak özelliklere sahip gruplandırılmış bir dizi olumsuz davranışı ifade etmek içindir

Çizelge 2.1. (devam) STIX bileşenleri [159]

Bileşen	Amaç
Araç (Tool)	Tehdit aktörleri tarafından saldırıları gerçekleştirmek için kullanılabilecek yazılımları paylaşmak için kullanılır
Açıklık (Vulnerability)	Bir sisteme erişmek için saldırgan tarafından kullanılabilecek hataları tanımlamak için kullanılır
Kimlik (Identity)	Bilgi kaynağını ifade etmek amacıyla birey, kuruluş veya grupları temsil etmek için kullanılır

VERIS standardı ilk versiyonlarında XML kullanmasına karşın STIX ve CYBOX gibi JSON veri formatını tercih etmiştir [162]. Resim 2.5 üzerinde VERIS standardı ile üretilmiş STİ gösterilmiştir.



```
{
  "action": {
    "misuse": {
      "variety": [ "Email misuse" ],
      "vector": [ "LAN access" ]
    } },
  "actor": { "internal": {
      "motive": [ "Convenience" ],
      "variety": [ "End user" ]
    } },
  "asset": { "assets": [
      { "variety": "U - Desktop" }
    ], "ownership": "Victim"
  },
  "attribute": {
    "confidentiality": { "data_disclosure": "Potentially" }
  },
  "reference": "http:// ",
  "victim": [
    {
      "country": "US", "employee_count": "100 to 1000",
      "state": "NY",
      "victim_id": "Plaza"
    } ]
}
```


Resim 2.5. VERIS ile üretilmiş STİ [142]

2.4.3. Siber tehdit istihbaratının önemi

STİ ile mevcut ve yeni saldırılar hakkında bilgi sahibi olunabilir. Böylece hakkında bilgi elde edilen saldırılar için korunma yöntemleri geliştirme ve önlem alma gibi kritik adımlar atılabilir. Dolayısı ile STİ'ler kullanılarak saldırılar henüz gerçekleşmeden dahi engellenebilir.

Standartlar kullanılarak oluşturulmuş STİ ile mevcut sistemlere ilgili bilgilerin aktarılması otomatik olarak sağlanabilir. Böylece STİ insan etkileşimine ihtiyaç duyulmaksızın ilgili güvenlik sistemlerine gönderilerek hakkında bilgiler bulunan saldırılar gerçekleşmeden engelleyici önlemler alınabilir. Bahruz Jabiyev ve Necati Ersen Şişeci tarafından ele alınan

bir çalışmada STIX/CYBOX standartları kullanılarak oluşturulmuş STİ'lerden Suricata için otomatik olarak kural üretimi sağlanmıştır [155].

Kurum ve kuruluş yöneticilerine hedef durumunda oldukları ve gerçekleşmek üzere olan saldırılar hakkında istihbarat sağlanarak siber tehdit faaliyete geçene kadarki süre içinde kritik varlıkların ve iş süreçlerinin korunması, ilgili personelin bilgilendirilmesi ve iş planının gerektiği gibi yapılması sağlanabilir.

STİ ile saldırı eğilimleri hakkında bilgi toplanabilir. Böylece kurum ve kuruluşlar ihtiyaç duyabileceği insan kaynağını ve gereken ekipmanları tedarik edebilir.

2.4.4. Siber tehdit istihbaratı paylaşmak için platformlar

STİ paylaşımı için kullanılan platformlara MISP ve IBM X-Force Exchange örnek gösterilebilir [7]. Bu platformlar sayesinde üretilmiş STİ diğer kullanıcıların faydalanabilmesi için paylaşılabilmekte ve/veya diğer kullanıcılar tarafından oluşturulmuş STİ elde edilebilmektedir.

IBM X-Force Exchange ile STİ'ler bulut üzerinde depolanmaktadır [7, 163]. Bu sayede ilgili STİ her yerden ve her zaman ulaşılabilir olması sağlanmaktadır. Ayrıca bulut üzerinde depolama ile STİ'lerin daha çok istemci ile paylaşılabilmesi mümkün kılınmaktadır. Tüm bunlara ek olarak ilgili platform paylaşılmış STİ'lerin herkese açık veya gizli olarak tanımlanabilmesine olanak sağlamaktadır.
Kamu kurumlarının veya şirketlerin sahip olduğu bilgileri herkese açık platformlar üzerinde tutması her zaman için mümkün değildir. Gizli olabilecek verilerin farklı şirketler üzerinde depolanması da istenilen bir durum değildir. Dolayısı ile kamu kurumları ve şirketler STİ'lerini kendi sistemleri üzerinde depolamak isteyebilir. Buna ek olarak bu depolanan verileri yalnızca belirli aktörler ile paylaşmak amaçlanabilir. İşte böyle bir ihtiyaç için MISP platformunun kullanımı uygundur. MISP bir sunucu olarak mevcut ağ içinde çalıştırılabilir [7, 164]. Dolayısı ile STİ'lerin kamu kurumları veya şirketler tarafından kendi sistemleri üzerinde depolanması sağlanabilir.

IBM X-Force Exchange ve MISP, STİ'nin ifade edilmesi ve standartlaştırılması için kullanılan STIX formatını desteklemektedir [7, 163, 164]. Ancak STİ taşıma protokolü olan TAXII yalnızca MISP ile kullanılabilmektedir.

2.4.5. Siber tehdit istihbaratı elde etmede yaşanan zorluklar

Siber güvenliğin sağlanmasında oldukça büyük öneme sahip olan STİ'nin üretilmesinde araştırmacılar bazı zorluklar ile karşılaşmaktadır. Bu zorluklar Çizelge 2.2 üzerinde gösterilmiştir.

Çizelge 2.2. STİ üretmede yaşanan zorluklar

Zorluk	Etki
Uzman bilgisine duyulan ihtiyaç	STİ elde edilememekte
	Yanlış değerlendirilmiş bilgiler içerebilmekte
Manuel işlemler içermesi	Uzun ve zahmetli çalışma sürecine ihtiyaç duyulması
	Fazla iş yükü gerektirdiğinden şirketlerin ve/veya kurumların STİ üretmek için zaman ayırmaması / ayıramaması
STS raporları arasında benzerlik olmaması	Kullanılan her bir STS raporunun veri modelinin iyi bilinmesine ihtiyaç duyulması
	Farklı STS raporları üzerinden STİ oluşturmanın güçlüğü

İlgili çizelge ile görüldüğü üzere yeterli uzman bilgisine ulaşılamadığı durumlarda STİ elde edilememekte veya yanlış değerlendirilmiş bilgiler içerebilmektedir. Uzman bilgisine sahip olunduğu durumlarda ise STİ üretme aşamalarının içerdiği manuel işlemlerden dolayı oldukça uzun ve zahmetli bir çalışma sürecine ihtiyaç duyulmaktadır. Ayrıca, saldırıların tespiti için kullanılan STS'ler tarafından üretilen raporlar arasında genel bir benzerlik olmayışı [28] ve bu sebeple her bir STS'nin çıktısının farklı veri modeli barındırmasından dolayı STİ üretmek için kullanılan her STS raporunun veri modelinin iyi bilinmesi gerekmektedir. Tüm bunlara ek olarak, STS'ler tarafından otomatik olarak üretilmediğinden dolayı kullanıcılar tarafından oluşturulmak zorunda olan STİ'nin üretilmesi için oldukça fazla iş yükü gerektiğinden dolayı şirketler ve/veya kurumlar tarafından zaman ayrılamamakta ya da ayrılan zaman gereksiz görülmektedir.

Tüm bu bilgiler ışığında, STİ oluşturmada kullanılan manuel adımların otomatize edilmesine ve uzman bilgisine gereksinim duyulmaksızın kolaylıkla STİ üretebilme yeteneğinin kullanıcılara kazandırılmasına ihtiyaç olduğu söylenebilir.

3. ÖNERİLEN SİSTEM TASARIMI

STİ üretmede yaşanan zorlukları en aza indirmek ve STİ üretmeyi kolaylaştırmak için bir sistem önerilmiştir. Çalışmada, ilgili sistemin tasarımı yapılmış, uygulaması geliştirilmiş ve geliştirilen uygulama ile örnek bir STİ üretilmiştir.

Bu bölümde ilk olarak, bir STS'de bulunması gereken bileşenler araştırılmıştır. Elde edilen bilgiler ışığında önerilen sistemin modüllerinin tasarımı yapılmıştır. Ayrıca, mevcut saldırı veri kümeleri incelenmiş ve önerilen sistemde kullanılan saldırı tespit sisteminin tasarımı için tercih edilebilecek veri kümeleri seçilmiştir.

3.1. Genel Sistem Tasarımı

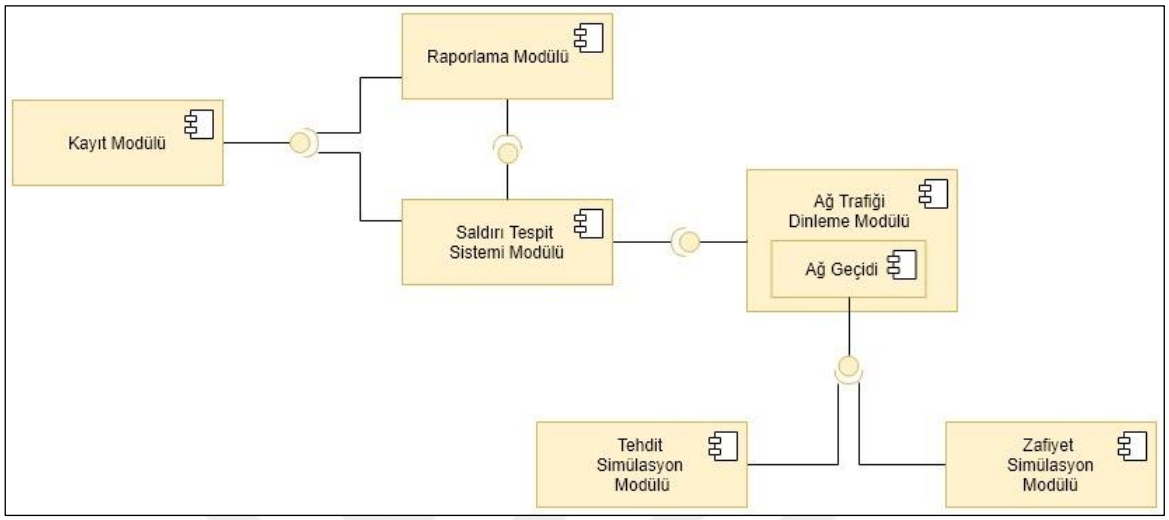
Literatür taraması doğrultusunda bir STS'de bulunması gereken bileşenler incelenmiştir. Bu kapsamda Herve Debar ve arkadaşları tarafından ele alınmış çalışmadan bir STS'de veri toplama bileşeni, veri formatlama bileşeni, analiz ve karar işlemleri için uzman sistem bileşeni ve güvenlik politikası bilgi tabanı bileşenine ihtiyaç duyulduğu görülmüştür [165]. Ayrıca, literatür taramasında basit bir STS tasarımının veri tabanı, konfigürasyon ve tespit edici bileşenleri barındırdığı bilgisine ulaşılmıştır [166].

Tüm bunlara ek olarak, Wenke Lee ve arkadaşları tarafından gerçekleştirilmiş bir çalışmada veri madenciliği temelli bir STS'de ham trafik bilgisinden formatlanmış kayıtlar elde eden sensör bileşeni, model oluşturma bileşeni, oluşturulmuş modeli kullanarak kayıtları sınıflandıran tespit edici bileşeni ve veri deposu bileşeni olduğu görülmüştür [167]. Literatür taramasında ayrıca bir STS'de karar verici mekanizmanın sonuçlarına göre rapor üretildiği bilgisine ulaşılmıştır [168].

Elde edilen bilgiler ışığında önerilen sistemin tasarımı; kayıt modülü, raporlama modülü, saldırı tespit sistemi modülü, ağ trafiği dinleme modülü, zafiyet simülasyon modülü ve tehdit simülasyon modülü olmak üzere 6 adet modül kullanılarak gerçekleştirilmiştir. İlgili modüller için bileşen diyagramı Şekil 3.1 üzerinde gösterilmiştir.

Önerilen sistem tasarımında bulunan ağ trafiği dinleme modülü ile kaydedilmiş trafik kayıtları veya canlı trafik kayıtları uygun formatta elde edilerek kayıt modülü üzerinde tutulur. Saldırı tespit sistemi modülü, STS fonksiyonları ile öğrenilmiş saldırı tiplerine göre ilgili kayıtları sınıflandırılır. Gerçekleştirilen sınıflandırma işlemini takiben saldırı olarak

etiketlenen trafik kayıtları kullanılarak raporlama modülü ile STİ oluşturulur. Tüm bunlara ek olarak ham verileri oluşturmak amacıyla tehdit simülasyon modülü ve zafiyet simülasyon modülleri kullanılır.



Şekil 3.1. Önerilen sistem için bileşen diyagramı

3.1.1. Kayıt modülü

Literatür taramasında karşılaşılan veri deposu, güvenlik politikası bilgi tabanı ve veri tabanı gibi bileşenlerin gerçeklenmesi için kayıt modülü tasarıma dahil edilmiştir. Kayıt modülü, içinde barındırabileceği veri tabanı ile STS'yi kullanabilmek ve STİ'yi üretebilmek için ihtiyaç duyulan kayıtları muhafaza etmek amacıyla kullanılır. Bu kapsamda; STS için saldırı veri kümeleri, veri madenciliğinde kullanılan parametreler, STİ üretmek için kullanılan gerçek trafik kayıtları gibi bilgiler ilgili modül üzerinde tutulur.

3.1.2. Raporlama modülü

Literatür taramasında karşılaşılan konfigürasyon ve karar verici mekanizmanın sonuçlarına göre rapor üretme gibi bileşenlerin gerçeklenmesi için raporlama modülü tasarıma dahil edilmiştir. Bu kapsamda raporlama modülü, raporlama fonksiyonlarının barındırılması ve modülün ihtiyaç duyabileceği yapılandırma ayarları için kullanılır. Dolayısı ile STİ üretme işlemi raporlama modülü ile gerçekleştirilir. Tüm bunlara ek olarak raporlama modülü; saldırı tespit sistemi modülüne erişmek, bilinmeyen saldırı kayıtları için uzman görüşü almak, elde edilmiş gerçek trafik kayıtlarını görüntülemek ve oluşturulmuş STİ'lere erişmek için kullanılır.

Yapılan araştırmalar sonucunda günümüzde en çok kullanılan STİ standardının STIX formatı olduğu görülmüştür [169]. Bu kapsamda STİ'nin, yaygın bir standart olan STIX formatında üretilmesi uygun görülmüştür.

3.1.3. Saldırı tespit sistemi modülü

Literatür taramasında karşılaşılan analiz ve karar işlemleri için uzman sistem, model oluşturma, oluşturulmuş modeli kullanarak kayıtları sınıflandıran tespit edici gibi bileşenlerin gerçeklenmesi için saldırı tespit sistemi modülü tasarıma dahil edilmiştir. Bu kapsamda saldırı tespit sistemi modülü, gerçek trafik kayıtlarının analiz edilmesi gibi saldırı tespit sistemi fonksiyonlarının yerine getirilmesi için kullanılır. Tüm bunlara ek olarak saldırı tespit sistemi; analizler neticesinde elde edilen bilinmeyen saldırıların uzman görüşü ile belirlendiği durumlarda öğrenme kümesini güncellemek, veri madenciliği işlemlerini gerçekleştirmek, ağ trafiği dinleme modülü ile uygun formatta elde edilmiş trafik kayıtlarıyla veri tabanını güncellemek gibi işlemleri yerine getirir.

Zeki yaklaşımların STS tasarımında kullanılmasının doğru bir seçim olduğu literatür taramasında görülmüştür [24]. Ayrıca, klasik imza tabanlı ya da anomali tabanlı STS'lerin performansını artırmak için veri madenciliğinden faydalanmış çalışmalar incelenmiştir [18, 23, 24]. Bu bilgiler ışığında ilgili STS, veri madenciliği teknikleri kullanılarak geliştirilir.

Subira ve Anitha tarafından gerçekleştirilmiş bir çalışmada veri madenciliği temelli bir STS içinde kullanılabilecek destek vektör makineleri, sinir ağları, bulanık mantık, bayes sınıflandırıcısı, k-en yakın komşu ve karar ağacı algoritmaları gibi algoritmalar karşılaştırılmıştır [25]. İlgili çalışmada aşağıda listelenmiş sonuçlara ulaşılmıştır:

- Destek vektör makineleri; yüksek karmaşıklık içermekte ve aşırı bellek kullanımına ihtiyaç duyulmakta
- Genetik algoritmaları; aynı problem için farklı yollardan çözüm bulabilmekte, ancak sabit bir süre içinde yanıt verememekte
- K-en yakın komşu algoritması; yüksek kayıt alanına ihtiyaç duymakta, öğrenme ve test süreleri uzun, veri boyutundan fazla etkilenmekte
- Sinir ağları; bağımlı ve bağımsız öznitelikler arasındaki doğrusal olmayan karmaşık ilişkilerin bulunmasında etkili, ancak yüksek hesaplama gücüne ihtiyaç duymakta ve öğrenme süresi uzun
- Bulanık mantık; bulanık sistemden bir model üretmenin zor olması

- Karar ağaçları; alan hakkında bilgi sahibi olunmasına ihtiyaç duyulmaması, yüksek boyutlu veri ile çalışabilmesi, gösteriminin kolayca anlaşılabilir olması, hem sayısal hem de kategorik veri ile çalışabilmesi

Elde edilmiş bilgiler ışığında, veri madenciliği temelli STS için veri madenciliği sınıflandırma tekniklerinden karar ağacı algoritmalarının kullanılmasının verimli olacağı görülmüştür. Bu kapsamda, ilgili veri kümeleri üzerinden saldırı tiplerini ve örüntülerini öğrenmek ve gerçek trafik kayıtlarını sınıflandırmak için karar ağacı algoritmaları kullanılır. Kullanılacak karar ağacı algoritmaları için gerçekleştirilen literatür taramasında ise ID3 ve CART algoritmalarının oldukça popüler ve sıklıkla tercih edilen karar ağacı algoritmaları oldukları görülmüştür [170, 171]. Bu sebeple ilgili STS de kullanılmaları uygundur.

3.1.4. Ağ trafiği dinleme modülü

Literatür taramasında karşılaşılan veri toplama, veri formatlama ve ham trafik bilgisinden formatlanmış kayıtlar elde eden sensör gibi bileşenlerin gerçeklenmesi için ağ trafiği dinleme modülü tasarıma dahil edilmiştir. Bu kapsamda ağ trafiği dinleme modülü, gerçek trafik kayıtlarını toplama ve bu kayıtları geliştirilmiş STS için uygun formatta elde etme amacıyla kullanılır. Trafik bilgilerinin çıkarılması ve ilgili trafik kaydına ait yeni öznitelik değerlerinin bu bilgiler üzerinden oluşturulması gibi işlemler ağ trafiği dinleme modülü ile gerçekleştirilir. Elde edilen trafik kayıtları, saldırı tespit sistemi modülü tarafından işlenmek üzere alınarak veri tabanına kaydedilir. Ayrıca, canlı trafik kayıtlarını işlemek amacıyla tüm trafiği üzerinden geçiren bir ağ geçidi ilgili modül üzerinde yapılandırılır. Bu kapsamda, tehdit simülasyon modülü ve zafiyet simülasyon modülü ağ trafiği dinleme modülü üzerinden haberleşir. Böylece ağ trafiği dinleme modülü, tehdit simülasyon modülü ve zafiyet simülasyon modülü arasındaki canlı trafiği inceleyebilir. Ağ trafiği dinleme modülü, aynı zamanda önceden kaydedilmiş gerçek trafik kayıtlarının da incelenmesine imkân sağlamalıdır. Dolayısı ile incelenmesi amaçlanan kaydedilmiş trafik ağ trafiği dinleme modülü üzerinde bulundurulmalıdır.

3.1.5. Zafiyet simülasyon modülü

Ham verileri oluşturmak amacıyla zafiyet simülasyon modülü ve tehdit simülasyon modülü tasarıma dahil edilmiştir. Zafiyet simülasyon modülü, saldırı yüzeyi oluşturma amacıyla kullanılır. İçinde zafiyetli servisler veya bal küpü sistemleri barındırabilir.

3.1.6. Tehdit simülasyon modülü

Ham verileri oluşturmak için tasarıma dahil edilen ve saldırı yüzeyi oluşturma amacıyla kullanılan zafiyet simülasyon modülü üzerinden saldırı kayıtları üretme amacıyla tehdit simülasyon modülü kullanılır. Bu kapsamda tehdit simülasyon modülü, saldırı araçları barındırabilir. Tehdit simülasyon modülü ve zafiyet simülasyon modülü ile önerilen sisteme bilinen veya bilinmeyen saldırı tipleri için trafik üretebilme kabiliyeti kazandırılır. Böylece sistem, hedeflenmiş saldırılar için STİ oluşturma yeteneği elde eder ve bu modüller üzerinden sistem tarafından bilinmeyen saldırıların öğrenilmesi sağlanır.

3.2. Saldırı Veri Kümeleri

Kullanılacak veri kümesi, tasarımı yapılan bir STS için oldukça büyük öneme sahiptir. İlgili veriler, STS'nin saldırıları öğrenmesi için eğitim kümesi olarak kullanılır. Ayrıca veri kümesinin bir bölümünden, geliştirilmiş STS'nin başarımını ölçmek için test amacıyla faydalanılır. Dolayısı ile kullanılan veri kümesi, geliştirilen STS'nin başarılı bir şekilde görevini yerine getirebilmesi için ve kabiliyetlerini ölçmek için oldukça etkilidir. Literatür taramasında; araştırmacıların, STS uygulamalarında kullanacakları veri kümesini seçerken genel kabul görmüş hazır veri kümeleri yerine kendi oluşturacakları veri kümesini seçmeyi tercih ettiklerinde oldukça zahmetli ve maliyetli bir çalışma gerçekleştirmenin zorluklarının ortaya çıkabileceği görülmüştür [24]. Bu sebeple, kullanılması muhtemel hazır veri kümeleri incelenmiştir.

3.2.1. Mevcut veri kümeleri üzerinde gerçekleştirilen incelemeler

Iman Sharafaldin ve arkadaşları tarafından 2016 yılında gerçekleştirilen bir çalışmada 1998 yılından itibaren araştırmacıların kullanımına sunulan 11 saldırı veri kümesi incelenmiştir [172]. Elde edilmiş sonuçlar şu şekildedir:

- *DARPA veri kümesi (1998):* Posta, gezinti, FTP, Telnet, IRC ve SNMP aktiviteleri barındırılmaktadır. Veri kümesi; servis dışı bırakma, parola tahmini, arabellek aşımı, uzak FTP, Syn saldırısı, Nmap ve Rootkit gibi saldırılar ile oluşturulmuştur. Ancak, gerçek dünya ağ trafiğini yeterince ifade edememektedir.
- *KDD CUP 99 veri kümesi (1999):* DARPA veri kümesinin güncellenmiş sürümüdür. Neptune, pod ve smurf gibi servis dışı bırakma saldırıları eklenmiştir. Ancak, birçok gereksiz kayıt bulunmaktadır. Ayrıca, çarpık test sonuçlarına yol açabilecek bozulmalar mevcuttur.

- *DEFCON veri kümesi (2000,2002):* Bayrak kapmaca olarak dilimize çevrilmiş Capture the Flag (CTF) yarışması sırasında gerçekleşen trafik üzerinden elde edildiğinden gerçek dünya ağ trafiğini yeterince ifade edememektedir.
- *CAIDA veri kümesi (2002-2007-2016):* 3 farklı veri kümesi bulunmaktadır. Her bir veri kümesi belirli saldırılara özgü oluşturulmuştur. Ayrıca, trafik kayıtlarının protokol bilgileri gibi verileri anonimleştirilmiştir. Barındırdıkları bu gibi eksikliklerden ötürü, STS kıyaslaması için kullanılabilecek etkili bir veri seti değildirler.
- *LBNL veri kümesi (2004,2005):* Orta ölçekli bir site üzerinden elde edilmiş ağ trafiği kullanılarak oluşturulmuştur. Ancak herhangi bir veri üzerinden bireysel IP bilgisine ulaşılmasını engellemek için ağır bir anonimleştirme işlemi uygulanmıştır.
- *CDX veri kümesi (2009):* Veri kümesinde barındırılan keşif ve saldırı trafiği Nikto, Nessus ve WebScarab gibi otomatize araçlar ile oluşturulmuştur. Ancak trafik çeşitliliği ve hacmi oldukça düşüktür.
- *Kyoto veri kümesi (2009):* Bal küpü sistemleri kullanılarak oluşturulmuştur. Ancak yalnızca bal küplerine yönlendirilen trafik kullanıldığından ağ trafiğinin sınırlı bir görüntüsüne sahiptir. Buna ek olarak normal trafik, simüle edilerek oluşturulduğundan ve yalnızca DNS ile posta trafiği kullanılarak üretildiğinden dolayı gerçek dünya ağ trafiğini yansıtmamaktadır.
- *Twente veri kümesi (2009):* OpenSSH, Apache Web Sunucusu ve Proftp gibi 3 servis kullanılarak bal küpü sistemleri vasıtası ile oluşturulmuştur. Saldırı çeşitliliği ve hacmi düşüktür.
- *UMASS veri kümesi (2011):* Tek bir TCP tabanlı saldırı senaryosu kullanılarak oluşturulmuştur. Dolayısı ile bir STS tasarımı için faydalı değildir. Trafik ve saldırı çeşitliliği oldukça düşüktür.
- *ISCX2012 veri kümesi (2012):* HTTP, SMTP, SSH, IMAP ve FTP protokolleri için ağ trafiği barındırılmaktadır. Ancak gerçek dünya istatistiklerine nazaran simüle edilmiş saldırı dağılımları benzer değildir. Dolayısı ile gerçek dünya ağ trafiğini yansıtamamaktadır.
- *ADFA veri kümesi (2013):* FTP ve SSH için kaba kuvvet saldırısı, Java meterpreter, Linux meterpreter, özel yetkilere sahip kullanıcı ekleme ve webshell saldırılarından oluşmaktadır. Ancak saldırı çeşitliliğinin düşük olduğu ve bazı saldırı davranışlarının normal trafikten belirgin bir şekilde ayrılmadığı görülmüştür.

Tüm bunlara ek olarak ilgili çalışmada, mevcut veri kümelerinde bulunan düşük veri hacmi, kısıtlı saldırı çeşitliliği, anonimleştirilmiş paket bilgileri ve düşük öznitelik kümesi gibi dezavantajları gidermek amacıyla CICIDS2017 veri kümesi oluşturulmuştur. Oluşturulan veri kümesi 8 dosya üzerinden sunulmaktadır. Her bir dosya, normal trafiğin yanı sıra farklı saldırı tiplerini barındırmaktadır. Veri kümesinde; Bot saldırısı, çeşitli dağıtık servis engelleme saldırıları, çeşitli servis engelleme saldırıları, Patator aracı ile gerçekleştirilen FTP ve SSH kaba kuvvet saldırıları, web saldırıları, port tarama saldırısı ve sızma saldırısı kullanılmıştır.

Ranjit Panigrahi ve Samarjeet Borah tarafından CICIDS2017 veri kümesi üzerinde 2018 yılında gerçekleştirilen incelemeler neticesinde aşağıda listelenmiş problemler tespit edilmiştir [173]:

- *Dağınık varlıklar:* Veri kümesinin ayrı 8 dosya ile sunulduğundan dolayı her bir dosya üzerinde işlem gerçekleştirmenin can sıkıcı bir iş olabileceği vurgulanmış ve tüm dosyaların birleştirilmesi önerilmiştir.
- *Yüksek hacimli veri:* Birleştirilen dosyalar ile 3 119 345 adet kayıt elde edilmiş ve bu durum veri kümesi hacminin oldukça büyük olduğunu ortaya koymuştur. Bu denli büyük veri kümesi ile işlem yapmanın zorluğundan bahsedilerek bu durum bir dezavantaj olarak tanımlanmıştır.
- *Eksik veri:* Birleştirilmiş veri kümesi incelendiğinde 288 602 adet kayıt için sınıf bilgisi bulunmadığı ve 203 adet kaydın eksik bilgiye sahip olduğu görülmüştür.
- *Yüksek sınıf dengesizliği:* Veri kümesinde saldırı tipleri arasında dengesizlik olduğu belirtilmiştir. Örneğin; birleştirilmiş veri kümesinde, 2 359 087 adet normal trafik bulunmasına karşın yalnızca 21 adet SQL enjeksiyonu saldırısı için kayıt vardır. Yüksek sınıf dengesizliği bulunan böyle bir veri kümesini kullanan STS'nin, sınıflandırma işleminde çoğunlukta olan sınıfa yönelenebileceği ve bu durumun yanlış alarm sayısı artırabileceği belirtilmiştir. Ayrıca, veri kümesinden rastgele örnek seçildiği durumlarda düşük sayıda kayıt içeren sınıfların örnekleme dahil edilmemesi muhtemel görülmüştür. İlgili madde, veri kümesinin sahip olduğu en büyük eksiklik olarak gösterilmiştir.

3.2.2. Seçilen veri kümeleri

Literatür taramasında; araştırmacılar tarafından 2009 ile 2018 yılları arasında STS geliştirmek için gerçekleştirilen çalışmalarda en sık kullanılan veri kümesinin KDD CUP

99 veri kümesi olduğu görülmüştür [174]. Buna ek olarak; KDD CUP 99 veri kümesinin, saldırı tespit sisteminde kullanılacak olan veri madenciliği algoritmalarının başarımını ölçmek ve karşılaştırabilmek için bir standart haline geldiği bilgisine de ulaşılmıştır [175, 176]. Bu sebeplerden dolayı; STS'nin tasarımı gerçekleştirebilmek, veri madenciliği algoritmalarının başarımlarını ölçmek ve bu başarımları literatürdeki benzer çalışmalar ile karşılaştırabilmek için KDD CUP 99 veri kümesinin kullanılması uygun görülmüştür.

KDD CUP 99 veri kümesi, KDD-99 Beşinci Uluslararası Bilgi Keşfi ve Veri Madenciliği Konferansı ile birlikte düzenlenen Üçüncü Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları Yarışması için kullanılan veri kümesidir. Bu yarışmada zararlı aktiviteleri ve zararsız aktiviteleri birbirinden ayırabilecek yeteneğe sahip bir saldırı tespit sistemi geliştirilmesi amaçlanmıştır. İlgili veriler ile askeri ağ ortamı trafik kayıtları simüle edilmiştir. Bu trafik kayıtlarında bulunan zararlı aktiviteler DOS, R2L, U2R ve Probing olmak üzere 4 ana gruba ayrılmaktadır. DOS kategorisi ile hizmet engelleme saldırıları temsil edilmektedir. R2L kategorisi ile uzak makineden yapılan yetkisiz erişim denemeleri ifade edilmiştir. U2R kategorisi ise ayrıcalık yükseltme saldırılarından oluşmaktadır. Keşif araçları ile yapılan tarama faaliyetleri ise Probing kategorisi altında gösterilmiştir. Kategoriler ve içinde bulundurdukları saldırı tipleri Çizelge 3.1 üzerinde listelenmiştir.

KDD CUP 99 veri kümesi, 41 adet öznitelik içermektedir. Bu öznitelikler ilgili trafikte kullanılan protokol, servis, bağlı kalma süresi, yanlış giriş sayısı, paketin büyüklüğü, son iki saniye içinde ilgili kaynak tarafından yapılmış bağlantı sayısı gibi birçok kritik bilgidir. Tüm bu bilgilere ek olarak ilgili trafiğin sınıf değeri de bulunmaktadır. Böylece 41 adet öznitelik ve 1 adet sınıf değeri üzerinde veri madenciliği teknikleri kullanılmaktadır.

Çizelge 3.1. KDD CUP 99 saldırı kategorileri ve barındırdıkları saldırılar

Grup Adı	**Barındırdıkları Saldırılar**
DOS grubu	back, land, neptune, pod, smurf, teardrop
R2L grubu	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
U2R grubu	buffer_overflow, loadmodule, perl, rootkit
Probing grubu	ipsweep, nmap, portsweep, satan

Literatür taramasında; KDD CUP 99 veri kümesinin kayıt sayısının oldukça fazla görüldüğü ve bu sebepten ötürü %10 KDD CUP 99 veri kümesinin hem eğitim hem de test işlemleri için kullanıldığı görülmüştür [177]. Ayrıca, araştırmacılar tarafından önerilmiş sistemlerin KDD CUP 99 veri kümesinin büyük veri hacminden dolayı zorluklar ile karşılaşabileceğine değinildiği ve %10 KDD CUP 99 veri kümesinin tercih edildiği bilgisine ulaşılmıştır [178]. Farklı çalışmalarda ise KDD CUP 99 veri hacminin yüksek olduğundan dolayı hesaplama maliyetini düşürmek için %10 KDD CUP 99 tercih edildiği gözlemlenmiştir [179]. Tüm bunlara ek olarak, örnek sayısının büyüdüğü durumlarda basit bilgisayarların başa çıkamayacağı büyüklükte hesaplama maliyetine gereksinim duyulabileceği ve bu sebepten ötürü %10 KDD CUP 99 tercih edildiği görülmüştür [180]. Bu bilgiler ışığında, %10 KDD CUP 99 veri kümesi tasarımı yapılan STS'nin algoritma başarımlarının ölçülmesi için tercih edilmesi uygun görülmüştür.

Benzer çalışmalarda, %10 KDD CUP 99 veri kümesi üzerinde hesaplanan algoritma başarımlarının 10 katmanlı çapraz doğrulama ile elde edildiği görülmüştür [181, 182]. Ayrıca, bazı çalışmalarda tabakalı 10 katlamalı çapraz doğrulama kullanılmıştır [183]. Literatürde araştırmasında, tabakalı 10 katlamalı çapraz doğrulamanın standart ve en gelişmiş doğrulama yöntemi olduğu bilgisi edinilmiştir [184]. Edinilen bilgiler ışığında, tabakalı 10 katlamalı çapraz doğrulamanın tercih edilmesi uygun görülmüştür. Tabakalı 10 katlamalı çapraz doğrulama ile 10 katlamalı çapraz doğrulama için oluşturulmuş her katın benzer oranda sınıf dağılımına sahip olması sağlanır [183].

Mahbod Tavallaee ve arkadaşları tarafından gerçekleştirilen çalışma ile KDD CUP 99 veri kümelerinin sahip olduğu dezavantajlar incelenmiş ve ilgili eksiklikleri gidermek amacıyla yeni bir veri kümesi olarak NSL-KDD veri kümesini araştırmacıların kullanımı için sunmuşlardır [185]. İlgili çalışmada, KDD CUP 99 veri kümesinin barındırdığı yanlış değerlere sahip 2 adet kayıt saptanmıştır. Ayrıca, KDD CUP 99 veri kümesinin yüksek sayıda gereksiz kayıt içerdiği gösterilerek ilgili veri kümesi üzerinde gerçekleştirilen analizlerde eğitim ve test veri kümeleri için kayıtların sırasıyla %78'nin ve %75'nin tekrarlanmış veriden oluştuğu bilgisine ulaşılmıştır.

Bahsi geçen çoklukta gereksiz kayıttan oluşan KDD CUP 99 veri kümesini kullanmanın aşağıda listelenmiş dezavantajlara sebep olacağı aktarılmıştır:

- Eğitim kümesinde bulunan gereksiz kayıtlardan dolayı, algoritmalar çok sayıda kayıt içeren sınıfa yönelebilmekte ve daha az sayıda bulunan sınıfları öğrenemeyebilmektedir.
- Test kümesinde bulunan gereksiz kayıtlardan dolayı, ilgili gereksiz kayıtları daha iyi sınıflandıran algoritmaları başarılı olarak göstermeye meyillidir.

Aynı çalışmada, KDD CUP 99 veri kümesi tarafından sunulan test kümesinin çarpık ve orantısız şekilde dağılım gösteren kayıtlar barındırdığı belirtilmiştir. Bu sebepten ötürü, önerilen sistemin başarımını ölçmek için KDD CUP 99 veri kümesi tarafından sunulan test kümesi kullanılmaması uygun görülmüştür.

İlgili çalışmada, KDD CUP 99 veri kümesinin dezavantajlarını gidermek amacıyla oluşturulmuş ve araştırmacılara sunulmuş NSL-KDD veri kümesinin KDD CUP 99 veri kümesinde bulunan gereksiz kayıtları silerek, tekrar eden kayıtlardan yalnızca bir kopya kullanılarak ve yanlış değerler içeren kayıtları veri kümesinden çıkararak elde edilen verilerden rastgele kayıtların seçilmesi ile oluşturulduğu belirtilmiştir. Tüm bunlara ek olarak, KDD CUP 99 veri kümesi ile kullanılan algoritmaların başarımlarının güvenilmez olduğuna ve bu başarımların sınıflandırıcıların kabiliyetlerini ölçmek için iyi bir gösterge olamayacağına değinilmiştir. Ayrıca NSL-KDD veri kümesinin, araştırmacılar tarafından 2009 ile 2018 yılları arasında STS geliştirmek için gerçekleştirilen çalışmalar incelendiğinde KDD CUP 99 veri kümesinden sonra sıklıkla tercih edilen veri kümelerinden olduğu sonucuna ulaşılmıştır [174]. Tüm bunlara ek olarak, 2016 yılında Kajal Rai ve arkadaşları tarafından gerçekleştirilen çalışmada NSL-KDD veri kümesinin en güncel veri kümesi olduğu belirtilmiştir [186]. Bu sebeplerden dolayı, tasarımı yapılan STS'nin gerçek trafik kayıtlarını sınıflandırmak için öğrenmesi gereken saldırı tiplerinin NSL-KDD Eğitim veri kümesi üzerinden elde edilmesi ve önerilen sistemin başarımlarının ilgili veri kümesi üzerinde de hesaplanması uygun görülmüştür. NSL-KDD Eğitim veri kümesinde bulunan saldırıların protokollere göre dağılımı Çizelge 3.2 üzerinde gösterilmiştir [187].

Literatürdeki benzer çalışmalar incelendiğinde, NSL-KDD veri kümesi kullanılarak hesaplanan başarımların NSL-KDD Eğitim ve NSL-KDD Test veri kümelerinin tamamı sırası ile eğitim ve test amacıyla kullanılabileceği görülmüştür [188, 189]. Ancak, NSL-KDD Eğitim veri kümesi üzerinde 10 katmanlı çapraz doğrulama gerçekleştirildiği bilgisine de ulaşılmıştır [190]. Bu bilgiler ışığında NSL-KDD veri kümesi ile tasarımı

yapılan STS'nin algoritma başarımlarını literatürdeki benzer çalışmalar ile kıyaslayabilmek için her iki yönteminde tercih edilmesi uygun görülmüştür.

Çizelge 3.2. NSL-KDD Eğitim - protokol düzeyinde saldırı kayıtları dağılımı

Protokol	**DOS Grubu**	**Probe Grubu**	**R2L Grubu**	**U2R Grubu**
ICMP	2847	4135	0	0
UDP	892	1664	0	3
TCP	42188	5857	995	49

Tüm bunlara ek olarak veri seçimi, önişleme, dönüştürme, veri madenciliği ve bilginin sunumu gibi seçilen veri kümelerine uygulanan veri tabanından bilgi keşfi süreçlerinde tercih edilen yöntemler önerilen sistemin uygulamasının geliştirilmesinde belirlenmiştir.

4. ÖNERİLEN SİSTEMİN UYGULAMASI

Tasarımı yapılmış kayıt modülü, raporlama modülü, saldırı tespit sistemi modülü, ağ trafiği dinleme modülü, zafiyet simülasyon modülü ve tehdit simülasyon modülü kullanılarak önerilen sistemin uygulaması geliştirilmiştir. İlgili modüller tasarımı yapıldığı gibi aşağıda belirtilen amaçlar için kullanılmıştır:

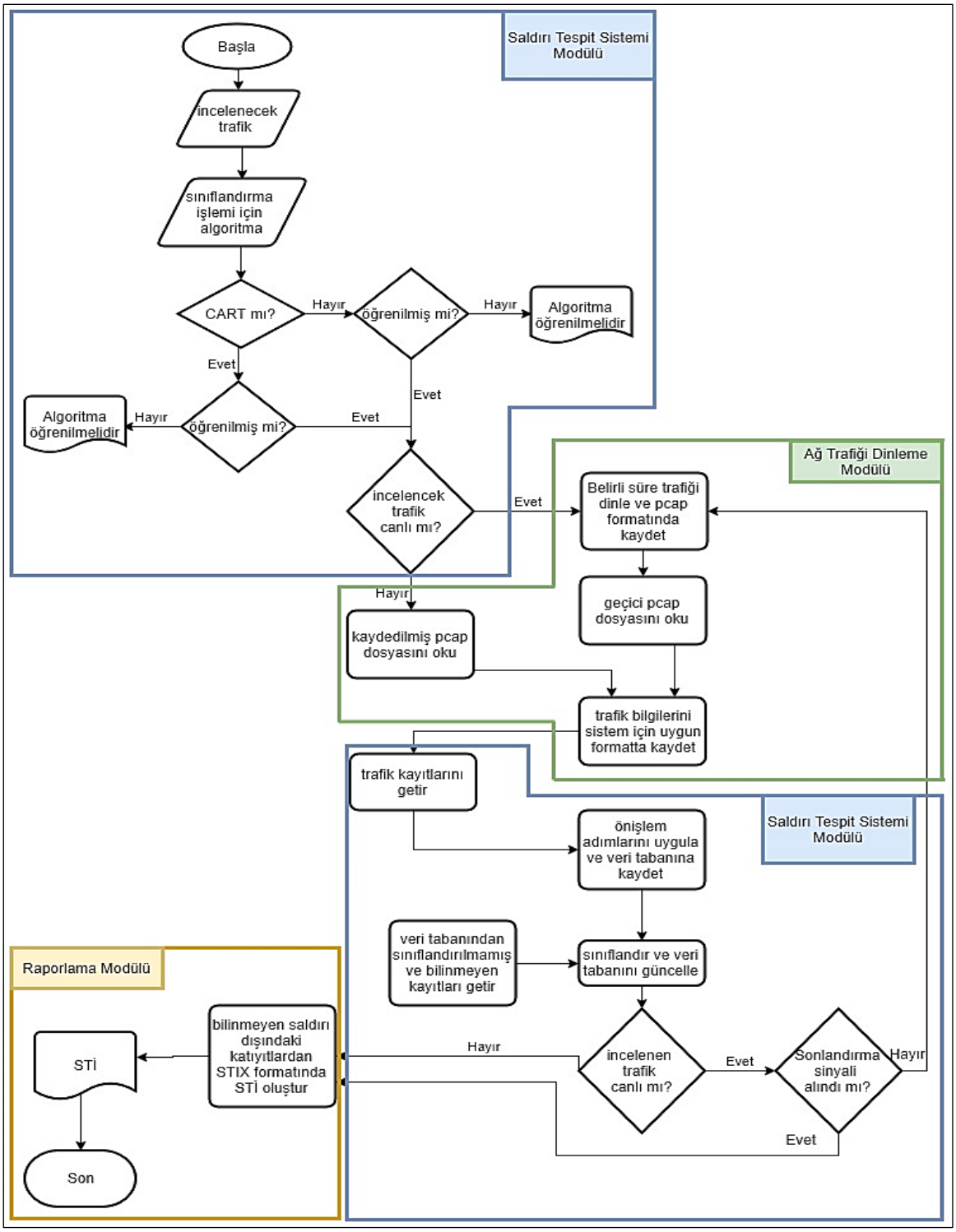
- *Kayıt modülü:* İlgili modül ile STS için saldırı veri kümeleri, veri madenciliğinde kullanılan parametreler, STİ üretmek için kullanılan gerçek trafik kayıtları gibi bilgiler tutulmuştur.
- *Raporlama modülü:* Saldırı tespit sistemi modülüne erişmek, bilinmeyen saldırı kayıtları için uzman görüşü almak, elde edilmiş gerçek trafik kayıtlarını görüntülemek, STİ üretmek ve oluşturulmuş STİ'lere erişmek için kullanılmıştır.
- *Saldırı tespit sistemi modülü:* Veri madenciliği adımları gerçekleştirilmiş ve STS işlevleri yerine getirilerek gerçek trafik kayıtlarını analiz edilmiştir. Ayrıca, analizler neticesinde elde edilen bilinmeyen saldırıların uzman görüşü ile belirlendiği durumlarda öğrenme kümesini güncellemek, ağ trafiği dinleme modülü ile uygun formatta elde edilmiş trafik kayıtlarıyla veri tabanını güncellemek gibi işlemleri yerine getirmiştir.
- *Ağ trafiği dinleme modülü:* Gerçek trafik kayıtlarını toplama ve bu bilgiler üzerinden geliştirilmiş STS için uygun formatta trafik kayıtları elde etme amacıyla kullanılmıştır.
- *Zafiyet simülasyon modülü:* Bal küpü sistemleri ile saldırı yüzeyi oluşturma amacıyla kullanılmıştır.
- *Tehdit simülasyon modülü*: Barındırdığı saldırı araçları ile zafiyet simülasyon modülünü kullanarak saldırı kayıtları üretmek için kullanılmıştır.

Geliştirilen uygulamada, bir veri madenciliği algoritmasının farklı uygulamalarda aynı sonucu vermeyebileceğinden ve farklı başarımlar gösterebileceğinden dolayı gerçek trafik kayıtlarını sınıflandırmak amacıyla kullanılacak algoritmanın seçilmesi için tasarımda belirlenen ID3 ve CART algoritmaları için kıyaslama yapılmıştır. İlgili kıyaslamalar, kullanılması muhtemel birçok veri kümesi arasından seçilmiş KDD CUP 99 ve NSL-KDD veri kümeleri için ayrı ayrı gerçekleştirilmiştir. Algoritmaların doğrulukları, öğrenme süreleri, sınıflandırma süreleri, yaprak sayıları, ağaç derinlikleri ile her bir saldırı tipi için hesaplanmış kesinlik, hassasiyet ve f1 skoru değerleri üzerinden kıyaslamalar yapılmıştır.

Yapılan kıyaslamalara ek olarak, literatürdeki benzer çalışmalar ile önerilen sistemin başarımlarını karşılaştırabilmek adına, %10 KDD CUP 99 veri kümesi ve NSL-KDD Eğitim veri kümesi üzerinde tabakalı 10 katlamalı çapraz doğrulama hesaplanmıştır. %10 KDD CUP 99 veri kümesi üzerinde önerilen sistem ile uygulanmış benzer adımlar veri madenciliğinde güvenilirliğini kanıtlamış Weka aracı üzerinde de gerçekleştirilmiştir. Weka aracı ile benzer sonuçlara ulaşıldığından ve en az literatürdeki benzer çalışmaların sahip olduğu başarımlar kadar sonuçlar elde edildiğinden önerilen sistemde uygulanmış veri madenciliği adımlarının ve geliştirilmiş algoritmaların başarılı olduğu kabul edilmiştir. Ancak, KDD CUP 99 ve %10 KDD CUP 99 veri kümelerinin barındırdığı gereksiz ve tekrar eden kayıtların bulunmasına ek olarak yanlış bilgilere sahip kayıtların mevcut olması gibi dezavantajlardan dolayı gerçek trafik kayıtlarının sınıflandırılmasında kullanılacak algoritmaların saldırıları öğrenmesi için NSL-KDD veri kümesi tercih edilmiştir [185]. NSL-KDD Eğitim veri kümesi üzerinde gerçekleştirilen tabakalı 10 katlamalı çapraz doğrulama hesaplanarak önerilen sistemin öğrenilen saldırı tipleri için başarımı önceki çalışmalar ile kıyaslanmıştır. Elde edilen sonuçların önceki çalışmalarda el edilmiş neticeler ile benzer olduğu görülmüştür. Ayrıca, NSL-KDD Eğitim veri kümesinin tamamı kullanılarak eğitilmiş algoritmalar ile NSL-KDD Test veri kümesi üzerinde yapılan sınıflandırmalar sonucunda elde edilen başarımın literatürdeki benzer çalışmalar ile kıyaslandığında önerilen sistemin benzer sonuçlara ulaştığı görülmüştür. Bu sebeplerden dolayı, önerilen sistemin veri madenciliği için başarılı bir şekilde kullanılabileceği ispatlanmıştır.

Geliştirilmiş uygulamada, saldırıları tespit edebilmek amacıyla yapılan sınıflandırma işlemi için hem kaydedilmiş trafik üzerinden elde edilen paketler incelenebilmekte hem de canlı trafikten yakalanan paketler üzerinde analiz yapılabilmektedir. Kullanıcı tarafından seçilen yönteme göre elde edilen trafik kayıtları istenilen algoritma ile sınıflandırılmaktadır. Gerçekleştirilen sınıflandırma işlemini takiben saldırı olarak etiketlenen trafik kayıtları kullanılarak STİ oluşturulabilir. Canlı veya kaydedilmiş gerçek trafik kayıtlarından STİ'nin elde edilmesi işlemi için akış diyagramı Resim 4.1 üzerinde gösterilmiştir. Geliştirilmiş uygulama ile örnek bir STİ üretmek için 24 saat boyunca internete açık çalıştırılan bir sunucuya yapılan saldırı trafiği kullanılmıştır. İlgili sunucu üzerine kurulan bal küpü sistemleri ile saldırganların dikkati çekilerek trafik kayıtları kaydedilmiştir. Kaydedilmiş trafik bilgisi, ağ trafiği dinleme modülü üzerine yerleştirilerek önerilen sistemin kaydedilmiş trafik kayıtlarını sınıflandırması sağlanmıştır. İlgili

sınıflandırma işlemi için ID3 algoritması kullanılmış ve NSL-KDD Eğitim veri kümesinin tamamı üzerinden öğrenme işlemi gerçekleştirilmiştir. Gerekli adımlar takip edilerek STIX formatında STİ elde edilmiştir. Böylece, önerilen sistemin STİ üretmek için başarılı bir şekilde kolaylıkla kullanılabileceği gösterilmiş ve internet ortamına açılacak olan sunucular için kullanılabilecek teknik STİ üretilmiştir.



Resim 4.1. Gerçek trafik kayıtlarından STİ elde edilmesi için akış diyagramı

4.1. Geliştirilen Modüller

Çizelge 3.1 üzerinde geliştirilen modüller için kullanılan sistem özellikleri listelenmiştir. Tüm modüllerin geliştirilmesi ve çalıştırılması için sanallaştırma teknolojilerinden faydalanılmıştır. Kayıt modülü, raporlama modülü, saldırı tespit sistemi modülü, ağ trafiği dinleme modülü, zafiyet simülasyon modülü ve tehdit simülasyon modülü Windows 10 işletim sistemi üzerinde kurulu olan Oracle VM VirtualBox sanallaştırma yazılımı ile oluşturulmuştur. Saldırı tespit sistemi modülü ve raporlama modülü aynı sunucu üzerinde kullanılmıştır.

Çizelge 4.1. Modüller ve sistem kaynakları

Modül	İşletim Sistemi	İşlemci	Bellek	Disk
Kayıt Modülü	Ubuntu 16.04 Server	1	2048 GB	100 GB
Raporlama Modülü	Ubuntu 16.04 Server	1	2048 GB	30 GB
Saldırı Tespit Sistemi Modülü				
Ağ Trafiği Dinleme Modülü	Ubuntu 18.04 Server	1	1024 GB	10 GB
Zafiyet Simülasyon Modülü	Ubuntu 18.04 Server	1	1024 GB	80 GB
Tehdit Simülasyon Modülü	Ubuntu 18.04 Server	1	1024 GB	5 GB

4.1.1. Kayıt modülü

Kayıt modülü ile STS'i kullanabilmek ve STİ üretebilmek için ihtiyaç duyulan kayıtları muhafaza etmek için veri tabanı yazılımı kullanılmıştır. Bu kapsamda ilgili modüle veri tabanı yönetim sistemi olarak MySQL 5.7 yazılımı kurulmuştur.

Veri madenciliği işlemleri belleğe alınan kayıtlar üzerinde yapıldığından veri tabanı performansı kritik bir öneme sahip değildir. Çünkü veri tabanı yalnızca yeni kayıtları eklemek, mevcut kayıtları sunmak veya güncellemek için kullanılmış olup herhangi bir fonksiyon, prosedür veya tetikleyici gibi veri tabanı sistemi bileşenlerinden faydalanılmamıştır. Dolayısı ile ilgili modül ve veri tabanı yönetim sistemi üzerinde kritik öneme sahip bir yapılandırma uygulanmamıştır. Yalnızca, ilgili modüle erişim ve yönetim amacıyla kullanılmış SSH servisi sunucu yazılımı kurulmuştur.

Oluşturulmuş veri tabanı şeması

Veri tabanı şeması adı ids_db olarak tanımlanmıştır. Bu şema altında 11 adet tablo oluşturulmuştur. Bunlar table_configs, table_discretization, table_kdd99_all, table_kdd99_sample, table_kdd99_test, table_real_traffic, table_expert, table_kdd99_10percent, table_nsl_train, table_nsl_test ve table_nsl_update tablolarıdır.

Veri tabanındaki table_configs tablosu ile her veri madenciliği adımının güncel durumu tutulur. Bu sayede her adımdan önce kontroller yapılabilir. Örneğin, eğitim kümesi oluşturmadan algoritma eğitilmesine izin vermemek için eğitim kümesinin oluşturulup oluşturulmadığı bu tablodan kontrol edilir ve mevcut duruma göre ilgili işlem iptal edilebilir. Tablo üzerinde parameter ve value olmak üzere iki sütun bulunmaktadır. Tabloda bulunan kayıtlardan sample_status ile örneklem verisinin oluşturulup oluşturulmadığı, data_format_status ile veri dönüşümü yapılıp yapılmadığı, test_status ile test kümesinin önişlem adımlarından geçip geçmediği, discretization_status ile sürekli verilerin kesikli veriye dönüştürülüp dönüştürülmediği ve algorithms_status ile algoritmaların veri kümesinden saldırı tiplerini öğrenip öğrenmediği bilgileri tutulmaktadır.

Veri tabanındaki table_discretization tablosu ile öğrenme kümesinde bulunan veriler için dinamik olarak hesaplanmış ve sürekli verilerin kesikli veriye dönüştürülmesi için kullanılmış ayırma noktaları tutulur. Bu ayırma noktaları, test veri kümesinin ve trafik kayıtlarının önişleminde de kullanılmaktadır. Tablo üzerinde parameter ve value olmak üzere iki sütun bulunmaktadır. parameter ile ilgili öznitelik değeri, value ile ise ayırma noktaları ifade edilir. Ayırma noktaları çok uzun metin değerler olarak tutulduğundan BLOB veri türü ile kaydedilmektedir.

Veri tabanındaki table_kdd99_all tablosu, veri kümesi olarak kullanılan KDD CUP 99 veri kümesi kayıtlarını tutmaktadır. Toplam 4 898 431 adet kayıt bulunmaktadır. Bu tablodan seçilen veriler üzerinden örneklem çıkarma ile eğitim ve test kümeleri oluşturulabilmektedir. table_kdd99_all tablosu üzerinde 41 öznitelikten ve 1 sınıf değerinden dolayı 42 sütun bulunmaktadır.

Veri tabanındaki table_kdd99_sample tablosu oluşturulmuş örneklem kümesi kayıtlarını, table_kdd99_test tablosu ise oluşturulmuş test kümesi kayıtlarını bulundurmaktadır. table_kdd99_sample tablosu sütunları ile table_kdd99_raw sütunları aynıdır.

table_kdd99_test tablosu ise bu tablolarda bulunan sütunlara ek olarak tahmin sütunu içermektedir.

Veri tabanındaki table_real_traffic tablosu, kaydedilmiş ya da canlı trafik üzerinden elde edilmiş kayıtları içermektedir. Yakalanan tüm trafik bilgileri bu tablo üzerinde tutulmaktadır. Veri madenciliği teknikleri ile öğrenilmiş model, bu tablo üzerinde kullanılarak trafik kayıtlarının sınıflandırılması sağlanmaktadır. table_kdd99_test tablosu ile aynı sütunlara sahiptir.

Veri tabanındaki table_ipv4_expert tablosu, ilgili algoritmalar tarafından bilinmeyen saldırı olarak sınıflandırılmış gerçek trafik kayıtlarını içermektedir. Geliştirilmiş STS, algoritmaların eğitimi sırasında bu tabloda bulunan kayıtları öğrenme kümesi kayıtları ile birleştirilerek güncel saldırıların da kullanılmasını sağlamaktadır. table_real_traffic tablosu ile aynı sütunlara sahiptir.

Veri tabanındaki table_kdd99_10percent tablosu, %10 KDD CUP 99 veri kümesi kayıtlarını tutmaktadır. Toplam 494 021 adet kayıt bulunmaktadır. Bu tablodan elde edilen veriler üzerinden tabakalı 10 katlamalı çapraz doğrulama gerçekleştirilmektedir. table_kdd99_all tablosu ile aynı sütunlara sahiptir.

Veri tabanındaki table_nsl_train tablosu ve table_nsl_test tablosu, sırası ile NSL-KDD Eğitim ve NSL-KDD Test veri kümelerine ait kayıtları bulundurmaktadır. İlgili tablolarda bulunan kayıtlar algoritmalar için eğitim ve test kümesi olarak kullanılmakta ve table_nsl_train tablosu kayıtları üzerinden tabakalı 10 katlamalı çapraz doğrulama gerçekleştirilmektedir. Her iki tablo, table_kdd99_all tablosu ile aynı sütunlara sahiptir. table_nsl_train tablosu ve table_nsl_test tablosu, sırası ile 125 973 adet ve 22 544 adet kayıt barındırmaktadır.

Veri tabanındaki table_nsl_update tablosu, NSL-KDD veri kümesinde bulunan saldırıların güncelliğini test etmek için kullanılan kayıtların geçici olarak veri tabanında tutulduğu ilgili tablodur. Bu tablo, saldırı tespit sistemi modülü ile kullanıcı istekleri doğrultusunda güncellenmektedir. Bu sebeple, sabit bir kayıt sayısı yoktur. Kullanıcı seçimine göre, ağ trafiği dinleme modülü üzerinde bulunan ilgili kaydedilmiş trafik kaydı üzerinden istenilen formatta elde edilen bilgiler bu tablo ile tutulur.

4.1.2. Raporlama modülü

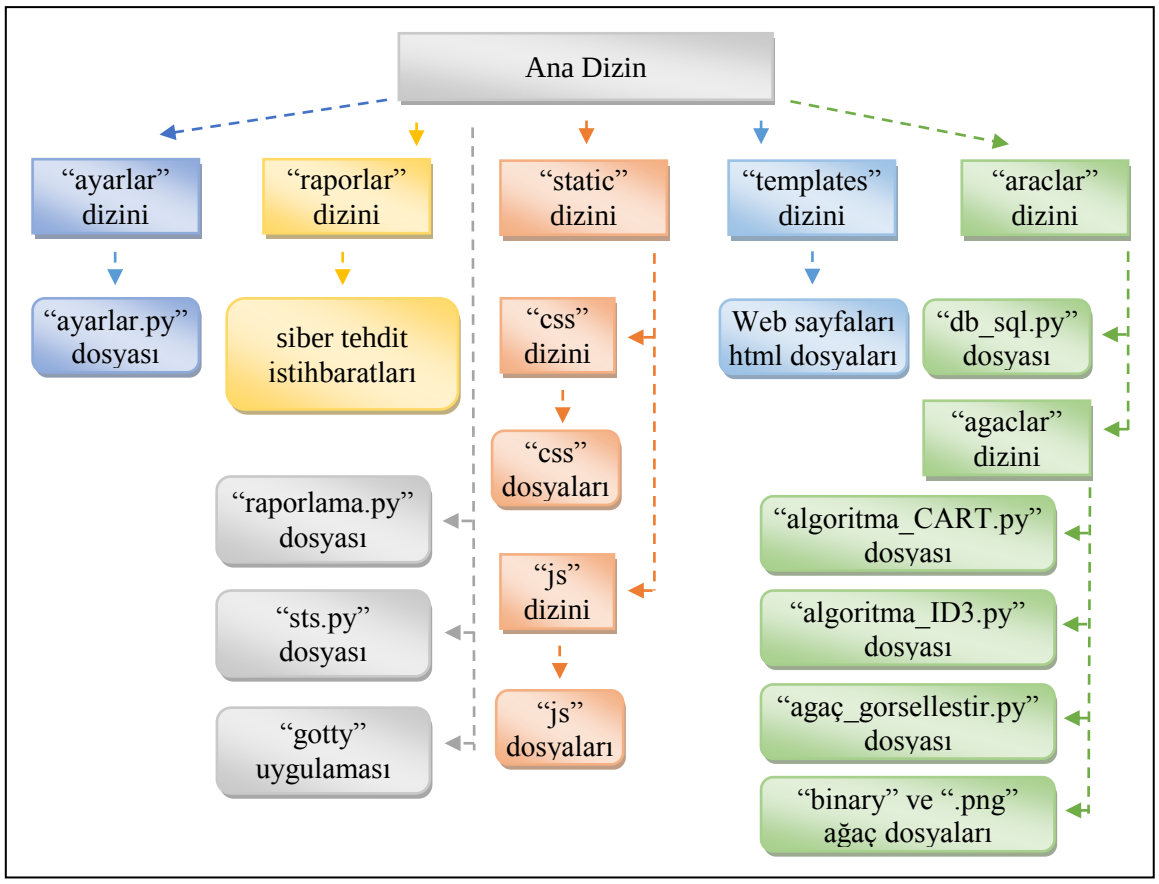
Saldırı tespit sistemi modülüne erişme, bilinmeyen saldırı kayıtları için uzman görüşü alma, elde edilmiş gerçek trafik kayıtlarını görüntüleme, STİ oluşturma ve oluşturulmuş STİ raporlarına ulaşma gibi işlemlerin yerine getirilebilmesi amacı ile raporlama modülü geliştirilmiştir. Raporlama modülü; anasayfa, saldırı tespit sistemi, bilinmeyen saldırı kayıtları, gerçek trafik kayıtları ve siber tehdit istihbaratları olmak üzere 5 adet sayfadan oluşmaktadır.

İlgili sayfaların ve işlevlerin geliştirilmesi için python3 programlama dili, zengin metin işaretleme dili (HTML) ve javascript betik dili kullanılmıştır. Ayrıca Flask, bootstrap, mysql-connector-python, flask_socketio ve stix2 gibi kütüphaneler kullanılmıştır. İlgili kütüphanelerin kullanım amaçları Çizelge 4.2 üzerinde gösterilmiştir. Tüm bunlara ek olarak, ilgili modüle erişim ve yönetim için SSH servisi sunucu yazılım kurulmuştur.

Çizelge 4.2. Raporlama modülü kullanılan kütüphaneler

Kütüphane	Kullanım Amacı
Flask	Python3 programlama dili kullanılarak web sayfalarının kullanıcılara sunulması sağlanmıştır.
Bootstrap	Sunulan web sayfalarında, ilgili kütüphane tarafından sağlanan düğme ve pop-up pencereler gibi hazır bileşenlerden yararlanılmıştır.
DataTables	Web sayfalarında kullanılmış tablolar için sayfalama ve tablo üzerinde arama gibi güçlü özellikler sağladığından kayıtların gösterilmesi için kullanılmıştır.
mysql-connector-python	Python3 programlama dili üzerinden mysql veri tabanı yönetim sistemine bağlanarak SQL komutlarının çalıştırılması sağlanmıştır.
flask_socketio	Raporlama modülünün anasayfası üzerinde gösterilen veri tabanına bağlantı için geçen sürenin asenkron şekilde iletilmesi için kullanılmıştır.
stix2	Python3 programlama dili üzerinden STIX formatında çıktı elde etmek için faydalanılmıştır.

Dosya ve Dizin Yapılandırması



Şekil 4.1. Uygulama dosya ve dizin yapısı

Raporlama modülü ve saldırı tespit sistemi modülü aynı sunucu üzerinde geliştirilmiştir. Geliştirilmiş uygulamaların dosya ve dizin yapısı Şekil 4.1 üzerinde gösterilmiştir. Ana dizin altında bulunan “raporlama.py” dosyası raporlama modülünün kaynak kodlarını içermektedir. “sts.py” dosyası ise saldırı tespit sistemi modülü kaynak kodlarını barındırmaktadır. “raporlama.py” dosyası vasıtasıyla gotty uygulaması üzerinden “sts.py” dosyası çalıştırılır ve saldırı tespit sistemi modülünün web hizmeti olarak sunulması sağlanır. Böylece, her türlü tarayıcı üzerinde geliştirilmiş saldırı tespit sistemi modülünün kullanımına olanak sağlanmıştır. Ancak, güvenlik zafiyeti oluşturmamak amacıyla gotty uygulaması üzerinden kullanıcı adı ve parola bilgileri kontrol edilmiştir. İlgili bilgiler sırası ile “suleyman” ve “arikan” olarak tanımlanmıştır. Geliştirilmiş saldırı tespit sistemi modülünün web hizmeti olarak yayınlanması için ilgili sunucunun 8080 portu kullanılmıştır.

“ayarlar” dizini raporlama modülünün ve saldırı tespit sistemi modülünün yapılandırma dosyasını içermektedir. Literatür taramasında, basit bir STS tasarımında konfigürasyon

bileşeninin bulunduğu görülmüştür [166]. Bu kapsamında, her iki modülün "ayarlar.py" dosyası ile yapılandırılması sağlanmıştır. "ayarlar.py" dosyasının güncellenmesi ile saldırı tespit sistemi modülü istenilen veri kümesi ile çalışmak için uygun hale getirilebilir. Bu dosya içinde veri tabanı erişim bilgeleri, tablo bilgileri, sütun bilgileri, modüllerin erişim bilgileri, ağaçların kaydedileceği dosya yolu ve kaydedilmiş trafiğin ağ trafiği dinleme modülü üzerindeki dosya yolu gibi birçok bilgi tutulmaktadır.

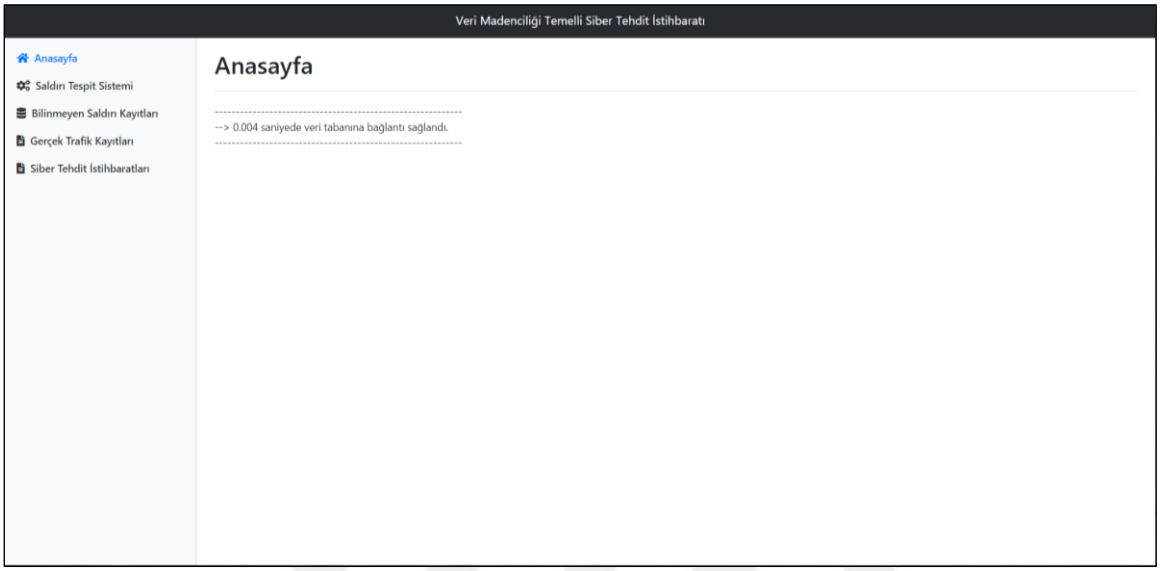
"raporlar" dizini üretilmiş STİ'leri bulundurmaktadır. Mevcut dizin ve dosyaların düzenli bir yapıda sunulabilmesi için elde edilmiş STİ'lerin raporlar dizini altında depolanması tercih edilmiştir. Düz yazı formatı kullanan dosyalar txt uzantısı ile bir STİ'nin ifade edilmesi için kullanılan STIX formatında üretilmiş dosyalar ise json uzantısı ile oluşturulmaktadır. Raporlama modülü vasıtası ile ilgili raporlar kullanıcıya sunulmaktadır. Raporların dosya isimleri için üretildikleri dosya ve tarih bilgileri kullanılmaktadır. STIX formatı kullanılarak üretilmiş bir STİ dosya adına 2018-11-25-20-30-30-644150.json örnek verilebilir.

"templates" ve "static" dizinleri, python3 programlama dili ile Flask kütüphanesinin kullanıldığı uygulamalarda sırası ile web sayfası şablonlarına ve ilgili web dosyalarına erişebilmek için kullanılan varsayılan dizinlerdir [192]. "static" dizini web sayfalarının stil dosyalarını bulunduran css dizinini ve javascript kodlarını barındıran js dizinini içermektedir. Böylece web sayfalarının ihtiyaç duyduğu harici dosyalar tek bir dizin altından sunulmaktadır. "templates" dizini ise web sayfaları için html uzantılı taslak dosyalarını içermektedir. Bu dosyalar python3 kütüphanesi olan Flask ile işlenerek web sayfaları üretilmektedir.

"araçlar" dizini MySQL veri tabanı ile bağlantı kurabilecek, kendisine verilen sorguları çalıştırarak ilgili metoda cevapları geri döndürebilecek ve büyük tabloların tahmini kayıt sayısını hesaplayabilecek metotları barındıran "db_sql.py" dosyasını ve agaclar dizinini içermektedir. "db_sql.py" dosyasındaki metotlar, geliştirilen uygulamaların birçok farklı parçasında ihtiyaç duyulduğundan ayrı bir dosyada kullanılmıştır. agaclar dizini altında ise veri madenciliği algoritmalarından ID3 ve CART algoritmalarının kaynak kodlarını barındıran sırası ile "algoritma_ID3.py" ve "algoritmaCART.py" dosyaları bulunmaktadır. Ayrıca öğrenilmiş bilgi ve örüntülerin bir ağaç yapısı olarak gösterilmesi için png formatında çıktı üreten "agaç_gorsellestirici.py" dosyası da buradadır. Tüm bunlara ek

olarak, öğrenilmiş ağaç bilgilerini içeren binary dosyalar ve png formatındaki çıktıları da agaclar dizininin altına kaydedilmektedir.

Raporlama modülü sayfaları



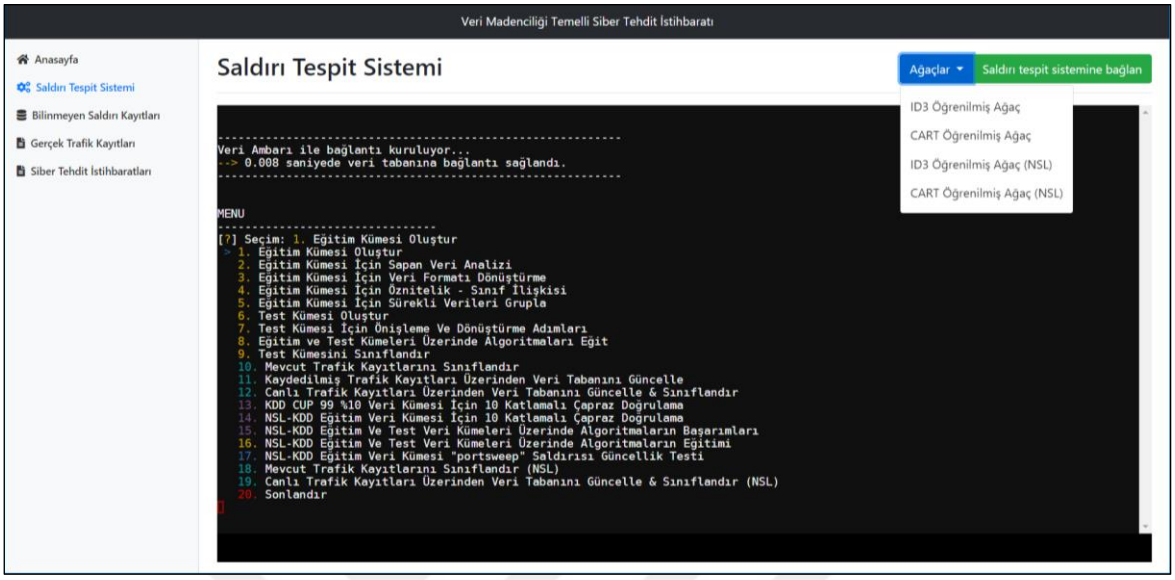
Resim 4.2. Raporlama modülü anasayfası

Resim 4.2 üzerinde raporlama modülü için anasayfa sayfası gösterilmiştir. Anasayfa üzerinde web soket kütüphaneleri kullanılarak asenkron haberleşme gerçekleştirilmiştir. Asenkron haberleşme ile raporlama modülünün veri tabanına ne kadar sürede bağlandığı bilgisi iletilir. İletilen bu bilgi, anasayfa üzerinde gösterilerek kullanıcıların veri tabanındaki kayıtların getirilmesi veya güncellenmesi hızı hakkında tahminde bulunabilmesi sağlanır. Uzun bağlantı süresi için veri tabanı üzerinde gerçekleştirilen ilgili işlemlerin hızı oldukça düşmektedir.

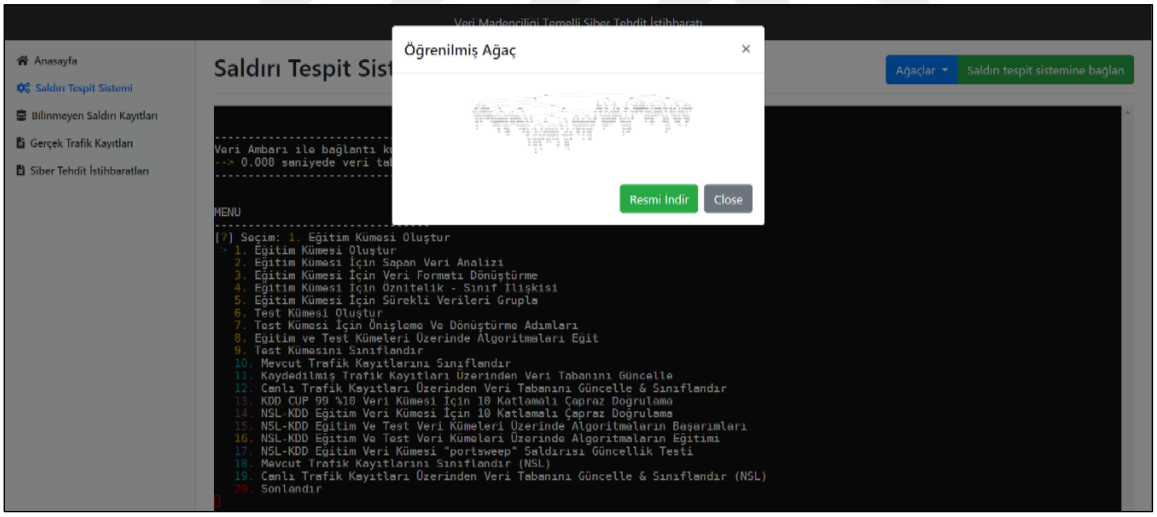
Resim 4.3 üzerinde saldırı tespit sistemi sayfası gösterilmektedir. Bu sayfa üzerinde bir HTML nesnesi olan iframe bulunmaktadır. “Saldırı tespit sistemine bağlan” düğmesine basılması ile kullanıcıdan kullanıcı adı ve parola bilgisi istenir. İlgili bilgiler, sırası ile “suleyman” ve “arikan” olarak tanımlanmıştır. Doğru bilgilerin girilmesinin ardından saldırı tespit sistemi modülü iframe nesnesi içinde kullanıcıya sunulur.

Saldırı tespit sistemi sayfasında bulunan “Ağaçlar” düğmesi üzerinden, veri madenciliği algoritmaları ile öğrenilmiş ve png formatında kaydedilmiş ağaçlara ulaşılabilir. Ağaçlar düğmesi vasıtası ile açılan seçeneklerden tercih edilen algoritma seçilerek ilgili resim

görüntülenebilir ve/veya indirilebilir. Resim 4.4 üzerinde NSL-KDD Eğitim veri kümesinden CART algoritması ile öğrenilmiş ağaç kullanıcıya sunulmaktadır.

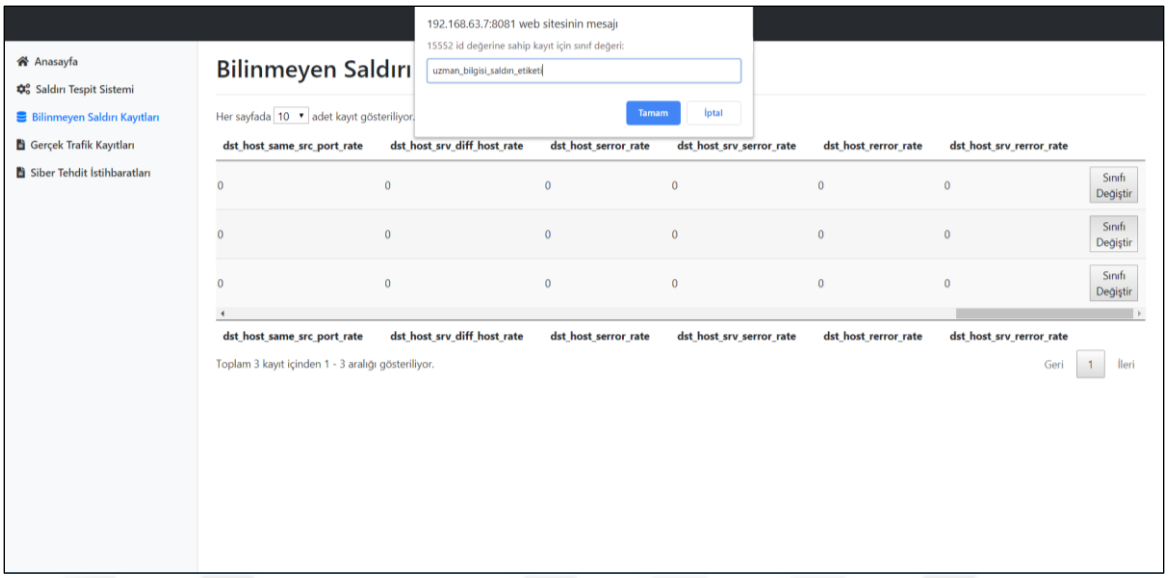


Resim 4.3. Saldırı tespit sistemi sayfası



Resim 4.4. Öğrenilmiş ağaca raporlama modülü üzerinden ulaşma

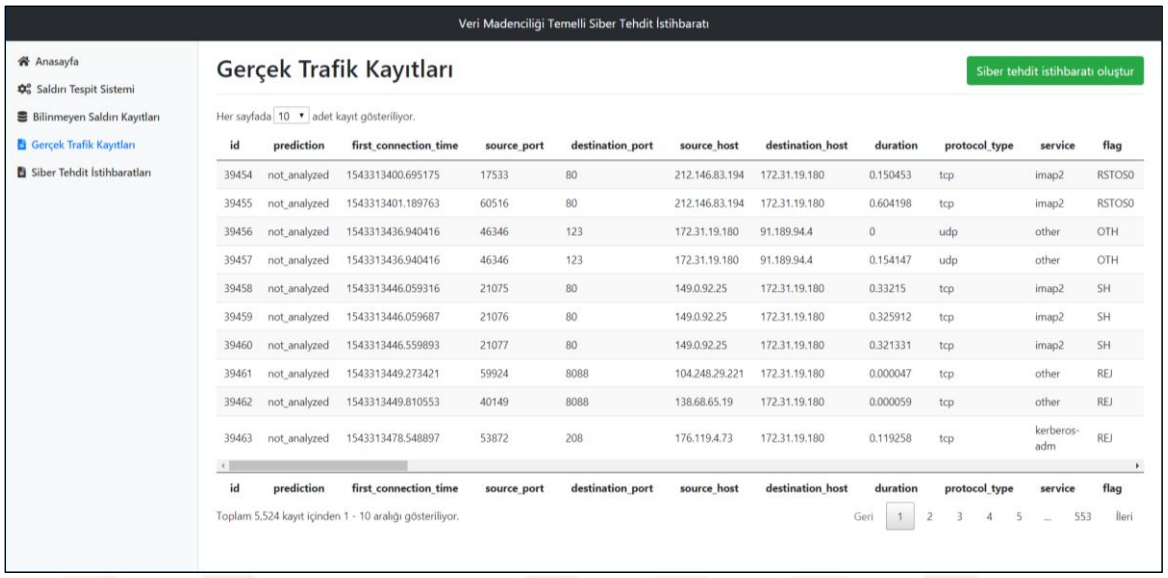
Bilinmeyen saldırı kayıtları sayfası üzerinde, veri madenciliği algoritmaları tarafından bilinmeyen saldırı olarak sınıflandırılmış gerçek trafik kayıtları bulunmaktadır. İlgili kayıtlar, uzman görüşü neticesinde tanımlanarak algoritmaların öğrenme kümesine dahil edilir. Sayfa üzerinde gösterilen her kayıt için "Sınıfı Değiştir" düğmesi bulunmakta ve ilgili düğme vasıtası ile açılan pop-up pencereye girilen sınıf bilgisi kaydedilebilmektedir. Resim 4.5 ile ilgili sayfa kullanılarak bir kayıt için uzman görüşü belirtme örneği gösterilmektedir.



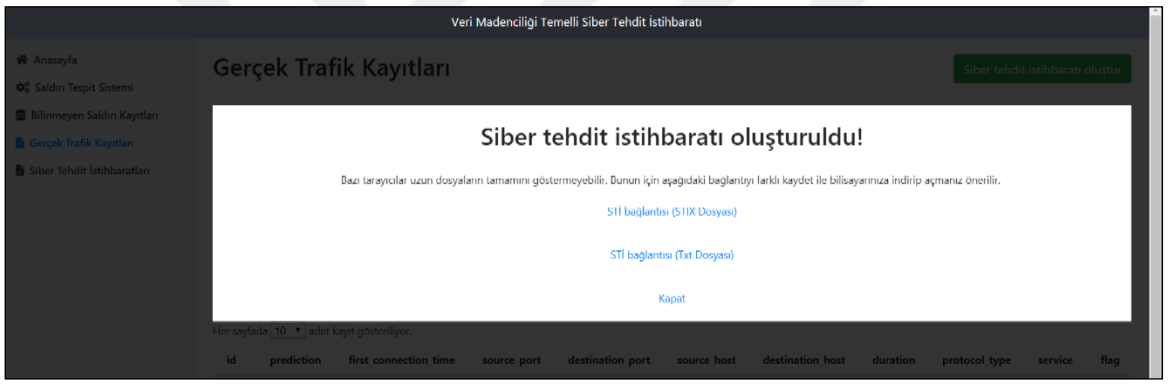
Resim 4.5. Bilinmeyen saldırı kayıtları sayfası

Gerçek trafik kayıtları sayfası ile canlı trafikten ve/veya kaydedilmiş trafikten elde edilen trafik bilgileri gösterilmektedir. Bu sayfa, ilgili tablonun mevcut kayıtlarını göstermektedir. Dolayısı ile trafik kayıtları üzerinde gerçekleştirilen veri madenciliği adımlarından dolayı gruplanmış veya değiştirilmiş veriler gözlemlenebilir. Tüm bunlara ek olarak, trafik kayıtları için veri madenciliği algoritmaları ile yapılan sınıflandırma sonuçları da ilgili sayfa üzerinden kullanıcıya sunulmaktadır. Resim 4.6 ile gerçek trafik kayıtları sayfası gösterilmiştir. İlgili ekran kaydı üzerinde, kayıtlara önişleme ve dönüştürme adımlarının uygulanmamış olduğu ve henüz sınıflandırılmadıkları görülmektedir.

Gerçek trafik kayıtları sayfası üzerinde bulunan “Siber tehdit istihbaratı oluştur” düğmesi ile STİ üretilmektedir. STİ oluşturmak için öncelikle tüm gerçek trafik kayıtları veri tabanından getirilir. Getirilen kayıtlardan sınıflandırma sonucu belli olanlar ve bilinmeyen saldırı olarak belirtilmemiş olanlar üzerinden STIX formatında ve düz yazı formatında STİ oluşturularak işletim sistemi üzerinde dosyalara kaydedilir. Kaydedilen dosya isimleri web sayfasına döndürülerek kullanıcı tarafından indirilmesi için bağlantı oluşturulur. Resim 4.7 üzerinde görüldüğü üzere oluşturulmuş STİ, hem json dosyası olarak STIX formatında hem de txt dosyası olarak düz metin formatında sunulmuştur.



Resim 4.6 Gerçek trafik kayıtları sayfası



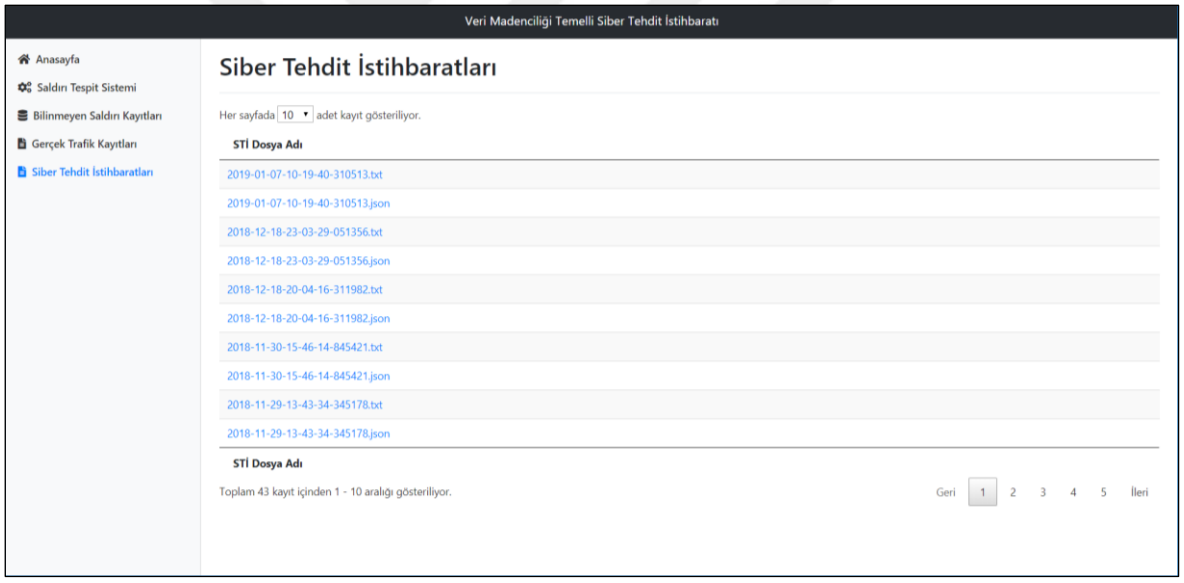
Resim 4.7. Üretilmiş STİ için sunulan formatlar

Resim 4.8 üzerinde üretilen iki farklı formattaki STİ gösterilmektedir. İncelenmiş trafik kayıtları, internet ortamına açılmış sunucu vasıtası ile kaydedilmiş gerçek trafik kayıtlarından elde edilmiştir. STIX formatında üretilmiş STİ; kimlik (identity), saldırı düzeni (attack pattern), gösterge (indicator) ve ilişki (relationship) nesnelerinden oluşmaktadır. İlgili nesneler ve kullanım amaçları ilgili başlık altında açıklanmıştır. Düz metin formatı ile üretilmiş STİ ise "-" ayracı ile bilgilerin birbirinden ayrıldığı formatta oluşturulmuştur. Sunulan bilgiler ilgili ekran görüntüsü üzerinden görülebilir.

Siber tehdit istihbaratları sayfası ile önceden oluşturulmuş STİ'lere ulaşılabilir. STİ dosya isimleri için tarih bilgisi kullanılmıştır. Resim 4.9 üzerinde görüldüğü gibi dosya isimleri "yıl-ay-gün-saat-dakika-saniye-mikrosaniye.uzantı" formatı ile oluşturulmuştur.



Resim 4.8. Farklı formatlarda üretilmiş STİ



Resim 4.9. Siber tehdit istihbaratları sayfası

4.1.3. Saldırı tespit sistemi modülü

Saldırı tespit sistemi modülü ile veri madenciliği adımları uygulanır ve STS işlevleri yerine getirilir. Bu kapsamda; eğitim kümesi oluşturma, sapan veri analizi, veri formatı dönüştürme, öznitelikler ile sınıf arasındaki ilişki katsayısını hesaplama, test kümesi oluşturma, test kümesi üzerinde önişleme ve dönüştürme adımları, algoritmaların öğrenmesi ve test kümesini sınıflandırma gibi işlemler gerçekleştirilmektedir. Tüm bunlara ek olarak gerçek trafik kayıtlarını pcap dosyasından okuyarak veya canlı trafik kayıtlarını yakalayarak veri tabanını güncelleme, mevcut gerçek trafik kayıtlarını sınıflandırma, %10

KDD CUP 99 ve NSL-KDD Eğitim veri kümeleri üzerinde katmanlı 10 katlamalı çapraz doğrulama gerçekleştirme, NSL-KDD Eğitim ve Test veri kümeleri üzerinde algoritmaların eğitimi ve NSL-KDD veri kümesi saldırı tiplerinin güncellik testi gibi işlemler de saldırı tespit sistemi modülü tarafından yapılmaktadır.

İlgili işlevlerin geliştirilmesi için python3 programlama dili kullanılmıştır. Tüm bunlara ek olarak, gotty uygulaması ile saldırı tespit sistemi modülünün bir web hizmeti olarak paylaşılması sağlanmış ve raporlama modülü içinden saldırı tespit sistemi modülüne erişim imkânı verilmiştir. Saldırı tespit sistemi modülünün geliştirilmesinde mysql-connector-python, numpy, scikit-learn, inquirer, ascii-graph, tqdm ve coloroma gibi kütüphaneler kullanılmıştır. İlgili kütüphanelerin kullanım amaçları Çizelge 3.2 üzerinde gösterilmiştir. Tüm bunlara ek olarak, ilgili modüle erişim ve yönetim için SSH servisi sunucu yazılım kurulmuştur.

Çizelge 4.3. Saldırı tespit sistemi modülü kullanılan kütüphaneler

Kütüphane	**Kullanım Amacı**
mysql-connector-python	Python3 programlama dili üzerinden mysql veri tabanı yönetim sistemine bağlanarak SQL komutlarının çalıştırılması sağlanmıştır.
numpy	Python3 programlama dili için güçlü dizi veri tipi sağladığından tercih edilmiştir.
scikit-learn	Öznitelikler ile sınıf değeri arasındaki ilişki kuvvetlerinin hesaplanmasında kullanılmış Extra Randomized Tree algoritması için faydalanılmıştır.
inquirer	Komut satırı üzerinde, ilgili kütüphane tarafından sağlanan seçim kutusu gibi hazır bileşenlerden faydalanılmıştır.
ascii-graph	Öznitelikler için hesaplanmış ilişki kuvvetlerini komut satırı üzerinde grafik ile göstermek için faydalanılmıştır.
tqdm	Komut satırı üzerinde ilerleme çubuğunun gösterimi için faydalanılmıştır.

Saldırı tespit sistemi modülü ve raporlama modülü aynı sunucu üzerinde geliştirildiğinden dosya ve dizin yapısı bir önceki bölümde sunulmuştur.

4.1.4. Ağ trafiği dinleme modülü

Ağ trafiği dinleme modülü, geliştirilmiş STS için uygun formatta gerçek trafik kayıtları elde etme amacıyla kullanılmıştır. Elde edilen trafik kayıtları, saldırı tespit sistemi modülü tarafından işlenmek üzere alınarak veri tabanına kaydedilmektedir. Canlı trafik kayıtlarını işlemek amacıyla tüm trafiği üzerinden geçiren bir ağ geçidi ilgili modül üzerinde yapılandırılmıştır. Bu kapsamda, tehdit simülasyon modülü ve zafiyet simülasyon modülü ağ trafiği dinleme modülü üzerinden haberleşmektedir. Böylece ağ trafiği dinleme modülü, tehdit simülasyon modülü ve zafiyet simülasyon modülü arasındaki canlı trafiği inceleyebilmektedir. Ağ trafiği dinleme modülü, aynı zamanda önceden kaydedilmiş gerçek trafik kayıtlarının da incelenmesine imkân sağlamaktadır. Dolayısı ile incelenmesi amaçlanan kaydedilmiş trafiğin bu sunucu üzerinde bulundurulması beklenmektedir.

Trafik bilgilerinin uygun formata dönüştürülmesi için bilgilerin çıkarılması ve ilgili trafik kaydına ait yeni öznitelik değerlerinin bu bilgiler üzerinden oluşturulması gibi işlemler ağ trafiği dinleme modülü ile gerçekleştirilmektedir. Önceden oluşturulmuş konfigürasyon dosyası ile kullanılan Bro IDS vasıtası ile kaydedilmiş trafik üzerinden belirli bilgiler elde edilir. Bu bilgilere paket kaynak IP adresi, hedef IP adresi, kaynak portu, hedef portu, zaman bilgisi örnek olarak gösterilebilir. C programlama dili ile yazılmış bir betik ile yeni öznitelik değerleri hesaplanır. Bu hesaplamalara son iki saniye içinde veya son 100 paket içinde ilgili kaynak tarafından yapılmış bağlantı sayısı bilgisinin bulunması işlemi örnek olarak verilebilir. Bu kapsamda kullanılan Bro IDS konfigürasyon dosyası ve C programlama dilinde yazılmış betik, literatür taraması ile elde edilmiştir.

Konfigürasyon dosyası ve betik, gurekddcup veri kümesinin oluşturulması için hazırlanmış çalışmalardır [193]. Gurekddcup veri kümesinin, KDD CUP 99 veri kümesi ile benzer karakteristik özellikler ile oluşturulması amaçlanmıştır [193]. İlgili veri kümesi, yapısında KDD CUP 99 veri kümesi özniteliklerini tümüyle barındırmasına ek olarak trafik paketlerinin payload bilgileri ile IP adresi ve port numarası gibi bağlantı bilgilerini de içermektedir. Ancak literatür taramasında, gurekddcup veri kümesinin barındırdığı verilerin %15'nin tekrar eden kayıtlar olduğu görülmüştür [194]. KDD CUP 99 veri kümesi öznitelikleri ile NSL-KDD veri kümesi öznitelikleri benzer olduğundan, bu çalışma kapsamında gurekddcup veri kümesinin oluşturulması için kullanılan Bro IDS konfigürasyon dosyası ve C programlama dilinde yazılmış betik ile pcap formatında bulunan trafik kayıtları NSL-KDD formatında elde edilmiştir.

Canlı trafik kayıtlarının elde edilmesi için tcpdump programından faydalanılmıştır. tcpdump programı kullanılarak belirli bir süre trafik kaydedilir. Kaydedilen trafik geçici olarak bir pcap dosyasında tutulur. İlgili pcap dosyası okunarak sistem için uygun formatta gerçek trafik kayıtları elde edilir. Elde edilen kayıtlar saldırı tespit sistemi modülü tarafından işlenmek üzere alınır. İlgili kayıtlar üzerinde önişleme, dönüştürme ve sınıflandırma adımları gerçekleştirilir. Tüm bu işlemlerin ardından henüz sonlanma sinyali alınmamış ise tcpdump programı ile trafiğin belirli bir süre kaydedilmesi aşamasına dönülür. Bu şekilde bir döngü sağlanarak canlı trafik kayıtları elde edilir. tcpdump programının trafiği kaydedeceği süre arttıkça kayıtlar üzerindeki doğru sınıflandırma sayısında da artış görülmektedir. Ancak, ilgili sürenin artması durumunda sistemin saldırılar için alarm üretmesi gecikmektedir. Buna ek olarak, büyük hacimdeki verinin tabloya eklenmesi ve sınıf etiketlerinin tahmin edilmesi yüksek işlem yükü oluşturmaktadır. İlgili sürenin 2 saniye altına kullanılması ise birçok saldırının tespit edilmesini engellemektedir. Çünkü C programlama dilinde yazılmış betik ile son 2 saniye üzerinden birçok istatistiki bilgi hesaplanmakta ve bu bilgiler öznitelik olarak kullanılmaktadır. Bu kapsamda, süre deneysel bir şekilde 10 saniye olarak belirlenmiştir.

4.1.5. Zafiyet simülasyon modülü

İçinde barındırdığı bal küpü sistemleri ile servisler simüle edilmektedir. Simüle edilen bu servislere 21 portunu kullanan ftp servisi, 22 portunu kullanan ssh servisi, 23 portunu kullanan telnet servisi, 80 portunu kullanan http servisi, 53 portunu kullanan dns servisi, 443 portunu kullanan https servisi, 5432 portunu kullanan postgresql servisi, 1433 portunu kullanan mssql servisi örnek olarak gösterilebilir. Tüm bu servislerin simüle edilebilmesi için birçok farklı düşük etkileşimli bal küpü sistemleri kullanılmıştır.

Birçok bal küpü yazılımını birlikte kullanmak için tek bir sunucu üzerine yapılacak kurulum işlemleri oldukça yüksek iş gücü gerektirebilmekte ve araştırmacıyı oldukça zorlu bir sürece itebilmektedir. Bu durumların önüne geçmek amacı ile içinde birçok bal küpü yazılımını barındıran honeydrive3 sanal makinesi tercih edilmiştir. Xubuntu Desktop 12.04.4 LTS işletim sistemi üzerinde önceden kurulmuş ve konfigürasyonları yapılmış 10 farklı bal küpü yazılımları ile birçok servis simüle edilebilmektedir [195]. Bu yazılımlara Kippo SSH honeypot, Dionaea, Amun Honeyd ve Glastopf örnek olarak gösterilebilir. Böylece FTP, SSH, TELNET, HTTP, HTTPS, POSTGRESQL ve MSSQL servisleri için

bal küpü yazılımları kolaylıkla kullanılmıştır. Tüm bu servislerin çalıştırılması ile bal küpü sunucusu üzerinde saldırı yüzeyi oluşturulmuştur.

4.1.6. Tehdit simülasyon modülü

Tehdit simülasyon modülü içinde; Nmap tarama aracı, Hydra kaba kuvvet saldırısı aracı ve Hping3 servis dışı bırakma saldırısı aracı gibi yazılımları içermektedir. Böylece, zafiyet simülasyon modülü üzerinde simüle edilen servisler kullanılarak saldırı trafiği üretilebilir.

4.2. Uygulanan Veri Tabanından Bilgi Keşfi Süreçleri

Seçilen veri kümeleri üzerinde; örneklem verisi oluşturma ile eğitim ve test kümlerinin elde edilmesi, sapan veri analizi, veri formatı dönüştürme, sınıf özniteliği ile diğer öznitelikler arasındaki ilişki kuvvetinin hesaplanması, öznitelik seçimi, sürekli verileri gruplama, algoritmaların eğitilmesi ile kayıtların sınıflandırılması gibi veri madenciliği adımlarına ek olarak algoritmalar vasıtası ile elde edilen örüntülerin kazanılmış bilgiye dönüştürülmesi gibi faaliyetleri içeren veri tabanından bilgi keşfi süreçleri gerçekleştirilmiştir.

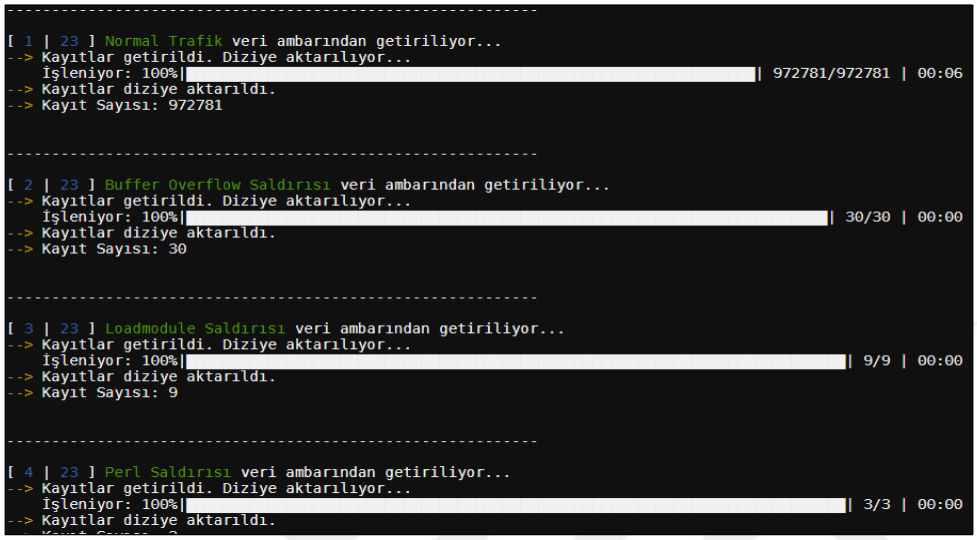
4.2.1. Veri seçimi adımı

Çalışmada, eğitim kümesini veya test kümesini oluşturmak için örneklem çıkarma işlemi uygulanmıştır. Örneklem oluşturma tekniklerinden katmanlı rastgele örnekleme ile örneklemde bulunan sınıfların frekansları ile tüm veri kümesinde bulanan sınıfların frekansları arasındaki oran, örneklem sayısı ile tüm veri kümesi sayısı arasındaki orana eşit tutulur. Böylece örneklemdeki sınıfların birbirine oranı, tüm veri kümesindeki sınıfların birbirine oranı ile aynı olur. Sonuç olarak, örneklemin tüm veri kümesi ile aynı özellikte olmasını sağlanır. Bu bilgiler ışığında, örneklem çıkarma işlemi için katmanlı rastgele örnekleme tercih edilmiştir. Tüm bunlara ek olarak, veri seçimi adımında sınıflandırılacak trafik kayıtlarının ağ trafiği dinleme modülünden getirilmesi ile veri tabanının güncellenmesi adımları da gerçekleştirilmiştir.

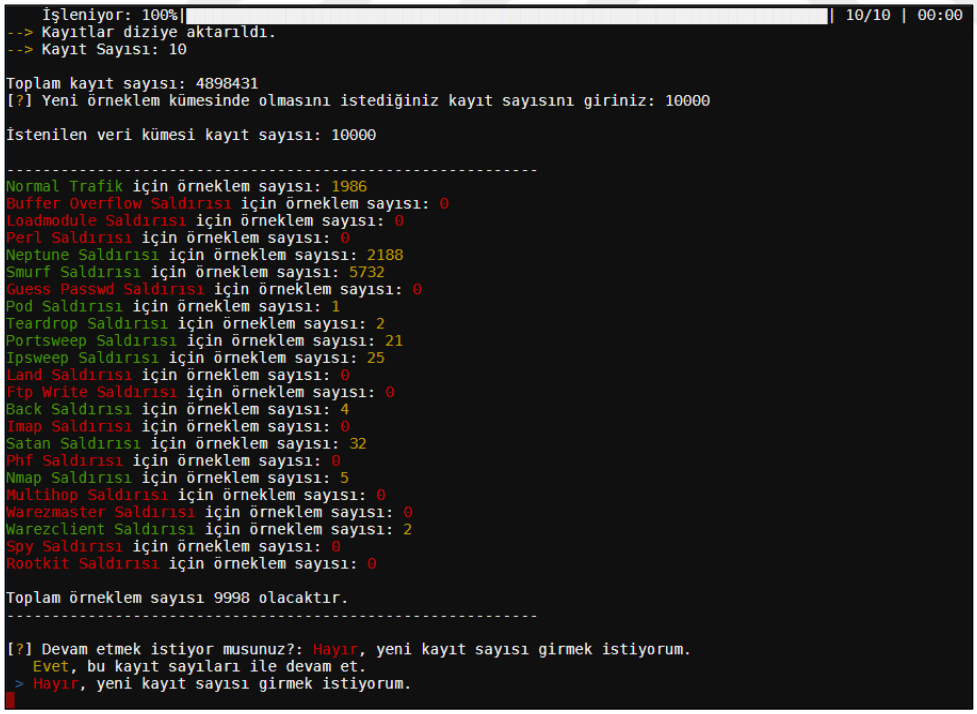
<u>Eğitim kümesi oluşturma işlemi</u>

Tüm kayıtlar veri tabanından getirilerek kullanıcıdan eğitim kümesinde bulunması istenen toplam kayıt sayısı bilgisi alınır. Tüm kayıtlardan ilgili sayıya oranla her sınıf için olması gereken kayıt sayıları hesaplanır ve kullanıcıya sunulur. Kullanıcı ilgili kayıt sayılarını

onayladığı takdirde örneklem çıkarma işlemine geçilir. Resim 4.10 üzerinde eğitim kümesi oluşturma işlemi için kayıtların veri tabanından getirilmesi aşaması ve Resim 4.11 üzerinde ise 10 000 kayıt ile oluşturulmak istenen eğitim kümesinde bulunacak sınıfların hesaplanan yeni kayıt sayılarının onaylanması aşaması gösterilmiştir.

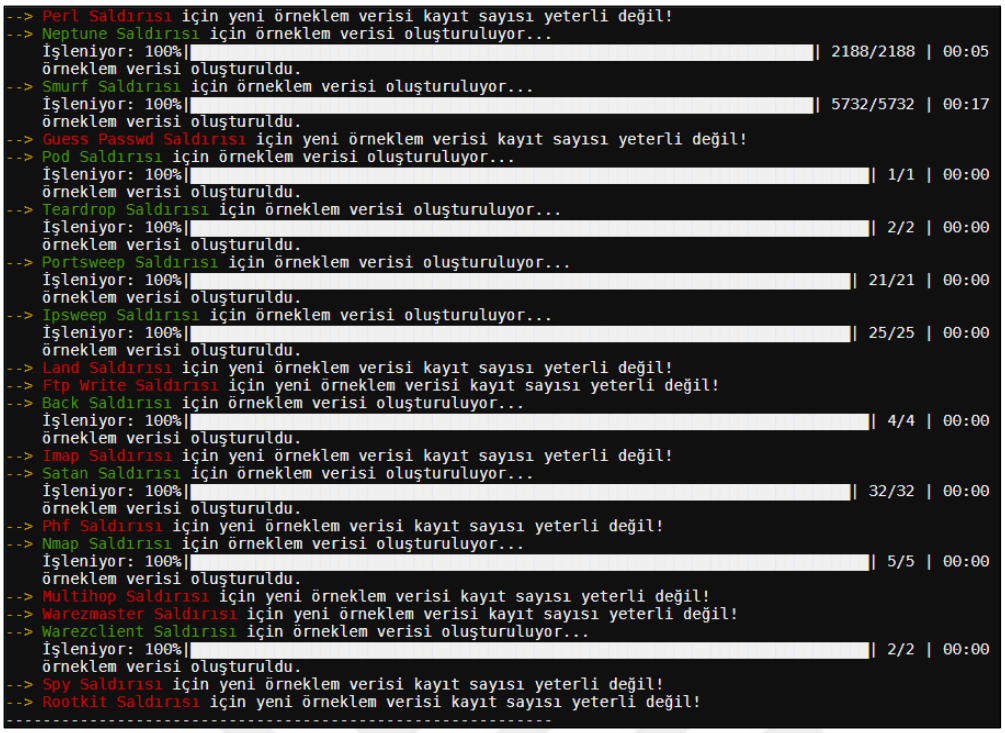


Resim 4.10. Eğitim kümesi oluşturma adımı - kayıtların getirilmesi



Resim 4.11. Eğitim kümesi oluşturma adımı - kayıt sayılarının onaylanması

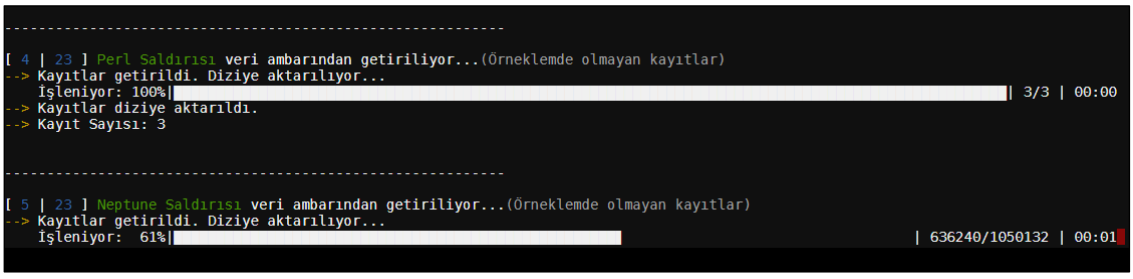
Hesaplanan kayıt sayılarının onaylanması ile Resim 4.12 üzerinde gösterildiği gibi örneklem çıkarma işlemi gerçekleştirilerek eğitim kümesi oluşturulur.



Resim 4.12. Eğitim kümesi oluşturma adımı - örneklem çıkarma

Test kümesi oluşturma işlemi

Test kümesi oluşturma işlemi, eğitim kümesi oluşturma işlemi ile aynı adımlara sahiptir. İlk olarak eğitim kümesinde olmayan tüm kayıtlar veri tabanından getirilerek belleğe alınır. Bellek üzerindeki kayıtlardan kullanıcıdan alınan yeni test kümesi toplam sayısı kullanılarak katmanlı rastgele örnekleme ile test kümesi oluşturulur. Resim 4.13 üzerinde test kümesi oluşturmak için kullanılacak olan kayıtların veri tabanından getirilmesi işlemi gösterilmiştir.

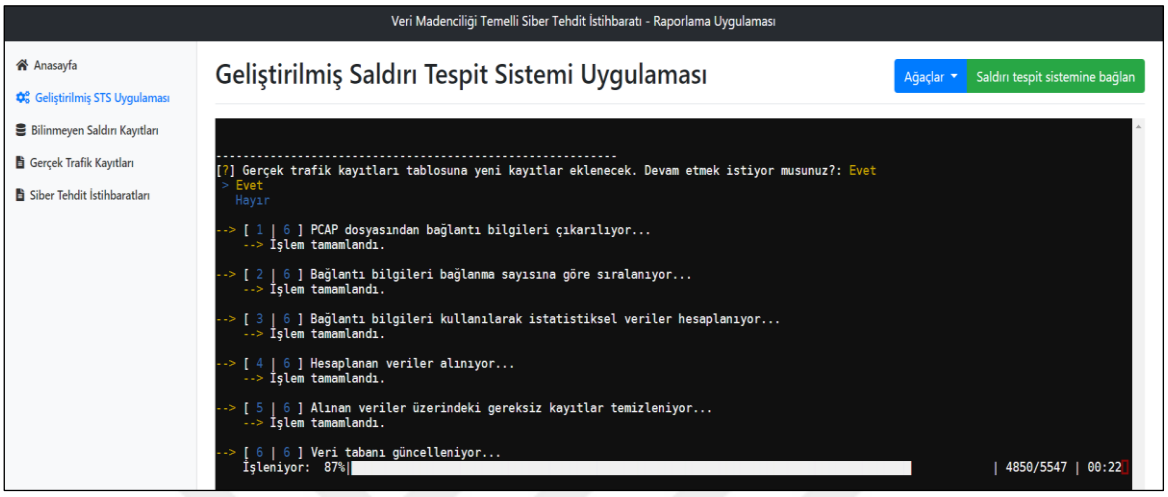


Resim 4.13. Test kümesi oluşturma adımı

Kaydedilmiş trafik kayıtları ile veri tabanının güncellenmesi işlemi

Kullanıcı, kaydedilmiş gerçek trafik kayıtları üzerinden trafik bilgilerinin çıkarılması ile yeni kayıtların ilgili tabloya ekleneceğine dair uyarılır ve onayı istenir. İlgili onayın

ardından, yapılandırma dosyasında dosya yolu belirtilmiş olan kaydedilmiş trafik dosyası okunarak veri tabanı güncellenir. Resim 4.14 üzerinde trafik kayıtlarının kaydedilmiş trafik üzerinden güncellenmesi ile ilişkili ekran görüntüsü gösterilmektedir.

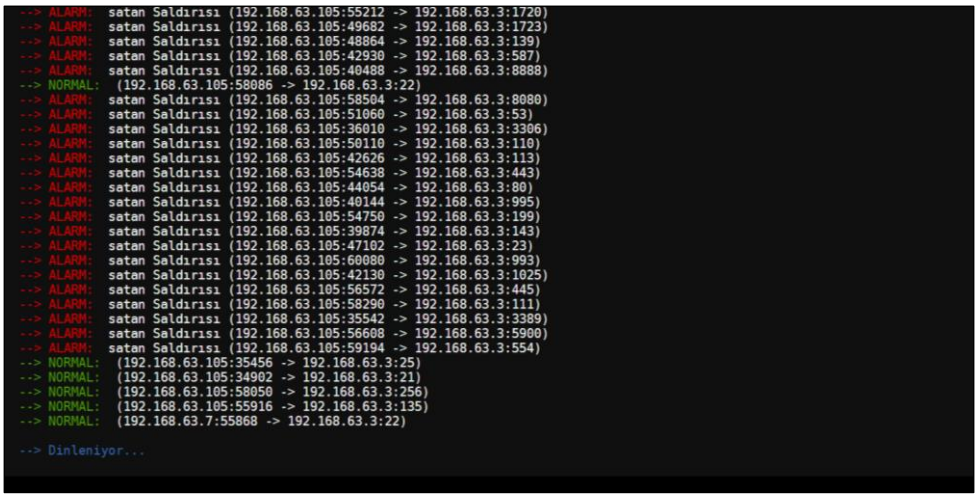


Resim 4.14. Kaydedilmiş trafik kayıtlarının veri tabanına kaydedilmesi adımı

Canlı trafik kayıtları ile veri tabanının güncellenmesi işlemi

Kullanıcının sınıflandırma işleminde kullanılacak olan veri madenciliği algoritmasını ve ilgili algoritma ile kullanılması istenen öznitelikleri seçmesi sağlanır. Ardından, yapılandırma dosyasında belirtilen süre kadar trafik kayıtları dinlenir. Dinleme süresi boyunca yakalanmış kayıtlar analiz edilerek kullanıcının tercihleri doğrultusunda çalıştırılan algoritma ile sınıflandırılır. Sınıflandırılmış ilgili kayıtlar ekran üzerinde gösterilir. Resim 4.15 üzerinde canlı trafik üzerinden yakalanmış Probing kategorisindeki saldırılar gösterilmektedir.

Tüm bunlara ek olarak, mevcut trafik kayıtlarının sınıflandırılması adımında olduğu gibi yapılan sınıflandırma işleminin neticesinde bilinmeyen saldırı olarak etiketlenen trafik kayıtları uzman görüşü için veri tabanında ilgili tabloya kopyalanır. Böylece, eğitim kümesinin yeni saldırı tipleri ile güncellenmesi sağlanır.



Resim 4.15. Canlı trafik üzerinden yakalanan kayıtlar

4.2.2. Önişleme adımı

Çalışmada, veriler üzerinde gerçekleştirilen önişleme adımında eksik veri analizi, sapan veri analizi ile gürültü verilerin temizlenmesi ve veri formatı dönüştürme işlemleri gerçekleştirilmiştir.

Eksik veri içeren kayıtlar, ilgili veri kümelerinin veri tabanına aktarılması esnasında kontrol edilmiştir. Kontrol işlemi, ilgili tablolar üzerinde çalıştırılmış SQL sorguları ile gerçekleştirilmiştir. Ancak KDD CUP 99, %10 KDD CUP 99 ve NSL-KDD veri kümeleri üzerinde yapılan incelemeler neticesinde eksik veri içeren kayıt gözlenmemiştir.

Literatür taramasında sapan veri analizi için çeyrekler açıklığı (interquartile range - IQR) kullanmanın basit olduğu, aykırı değerlere karşı düşük hassasiyet ile çalıştığı ve veri dağılımı hakkında kısıtlı bilgi ile dahi iyi sonuç alabildiği görülmüştür [196]. Bu sebeplerden dolayı sapan veri analizi için IQR hesaplaması kullanılmıştır. IQR hesaplaması 3 nokta üzerinden gerçekleştirilir. Bu noktalardan biri medyan iken diğerleri Q3 ve Q1 olan iki çeyrek noktasıdır. İlgili adımda öncelikle tüm veriler sıralanır. Sıralanan verilerin medyan değeri bulunur. Medyan değerinden küçük olan değerlerin medyanı Q1 ile ifade edilerek ilk çeyrek, medyandan büyük değerlerin medyanı ise Q3 ile ifade edilerek üçüncü çeyrek olarak tanımlanır. IQR değeri ise ilgili iki çeyreğin mutlak değer farkına eşittir [197]. IQR değerinin 1,5 katı ilk çeyrekten çıkarılarak alt sınır elde edilir. IQR değerinin 1,5 katı üçüncü çeyreğe eklenerek ise üst sınır elde edilmektedir. Alt sınırdan küçük olan veya üst sınırdan büyük olan veriler sapan veri olarak kabul edilir. Geliştirilen uygulamada, sapan veri olarak saptanmış kayıtlar ilgili örneklemden silinir.

Hem kategorik hem de sayısal veriler üzerinde veri madenciliği gerçekleştirebilmek için veri dönüştürme işlemi uygulanmıştır. İlgili veri kümesinde bulunan kategorik değerler kullanıcıların bildiği sayısal değerler ile değiştirilmiştir. Buna ek olarak, önişleme adımından geçen kayıtların kullanıcılar için hala anlaşılabilir olması için yapılandırma dosyasında tanımlanmış transform_to_dummy_data_process fonksiyonu kullanılmıştır. İlgili fonksiyon ile kayıtlar üzerinde gerçekleştirilen işlemler Çizelge 4.4 üzerinde gösterilmiştir.

Çizelge 4.4. Kategorik değerlerin sayısal değerler ile değiştirilmesi

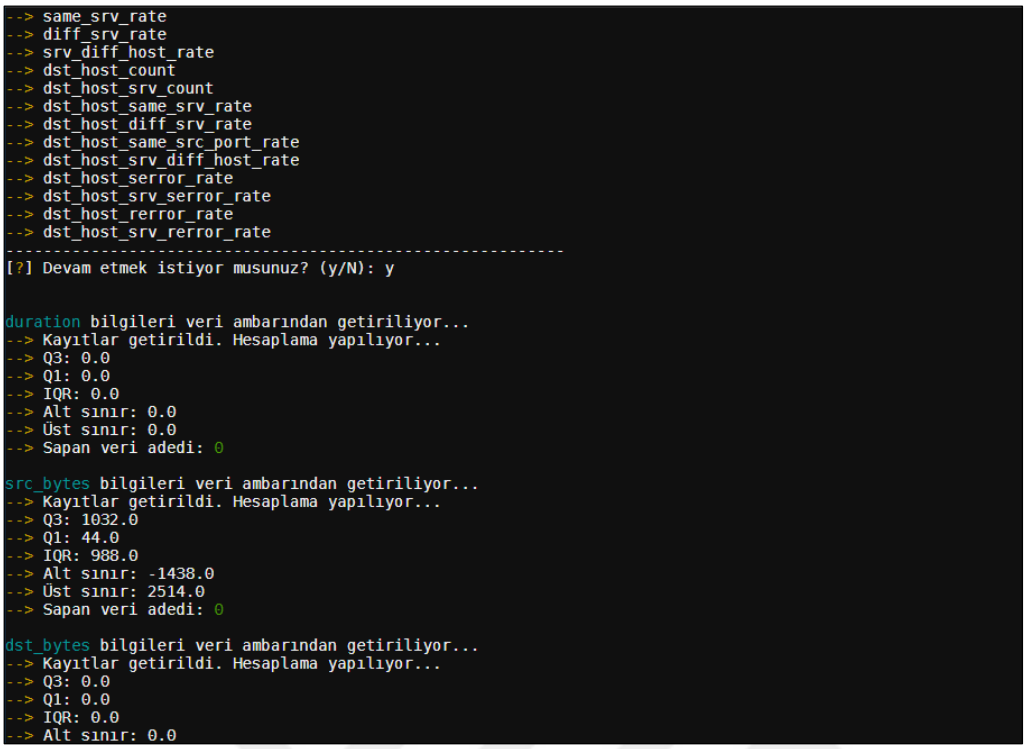
Öznitelik	**Kaynak - Kategorik Değerler**	**Hedef - Sayısal Değerler**
protocol_type	icmp, udp, tcp	1, 2, 3
flag	OTH, REJ, RSTO, RSTOS0, RSTR, RSTRH, S0, S1, S2, S3, SF, SH, SHR	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
service	other, private, tcpmux … tfido, fido	0, 1, 2 … 344, 345

Sapan veri analizi işlemi

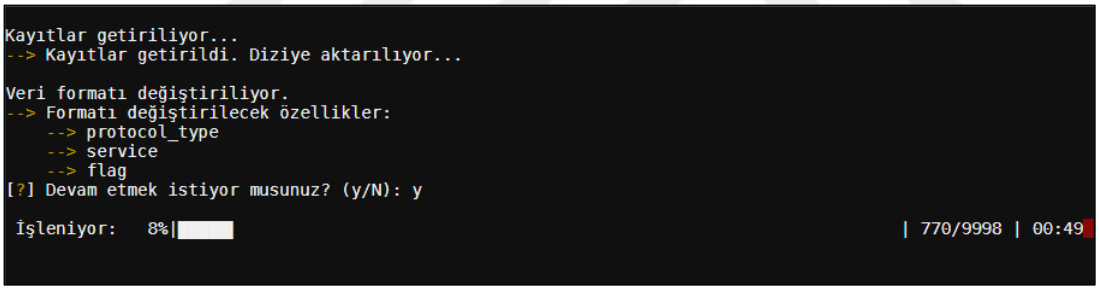
Kullanıcı, sapan veri kontrolü yapılacak öznitelikler hakkında bilgilendirilir. İlgili öznitelikler onaylandığı takdirde her öznitelik için ayrı IQR hesaplaması yapılarak kontrol gerçekleştirilir ve sapan veri sayısı görüntülenir. Bulunan sapan veriler veri kümesinden silinir. İlgili ekran görüntüsü Resim 4.16 ile gösterilmiştir.

Veri formatı dönüştürme işlemi

Veri formatı dönüştürme adımının gerçekleştirilmesi için ilgili işlemin uygulanacağı öznitelikler listelenerek kullanıcıdan ilgili işlemi onaylaması istenir. Kullanıcının işlemi onaylaması durumunda öncelikle tüm kayıtlar veri tabanından getirilir. Yapılandırma dosyasında tanımlanmış olan transform_to_dummy_data_process fonksiyonu kullanılarak kategorik değere sahip veriler belirtilen sayısal veriler ile değiştirilir. İlgili ekran görüntüsü Resim 4.17 ile gösterilmiştir.



Resim 4.16. Sapan veri analizi adımı



Resim 4.17. Veri formatı dönüştürme adımı

4.2.3. Dönüştürme adımı

Çalışmada, veriler üzerinde gerçekleştirilen dönüştürme adımında sürekli verilerin gruplandırılması, öznitelikler ile sınıf etiketi arasındaki ilişki kuvvetinin hesaplanması ve öznitelik seçimi işlemleri gerçekleştirilmiştir.

Sürekli değerler barındıran her öznitelik için ayrı ayırma noktaları dinamik olarak hesaplanmaktadır. Sürekli verinin ayrık veriye dönüştürülmesi için kullanılacak olan ayrıma noktalarının tespiti için kova metodu uygulanmıştır. Kova metodu tekniklerinden eşit genişlik metodunun çarpık veriyi iyi ifade edememesi ve sapan verilerden fazla etkilenmesine ek olarak eşit derinlik metodunun ölçeklenebilir olmasından dolayı eşit derinlik metodu tercih edilmiştir [198].

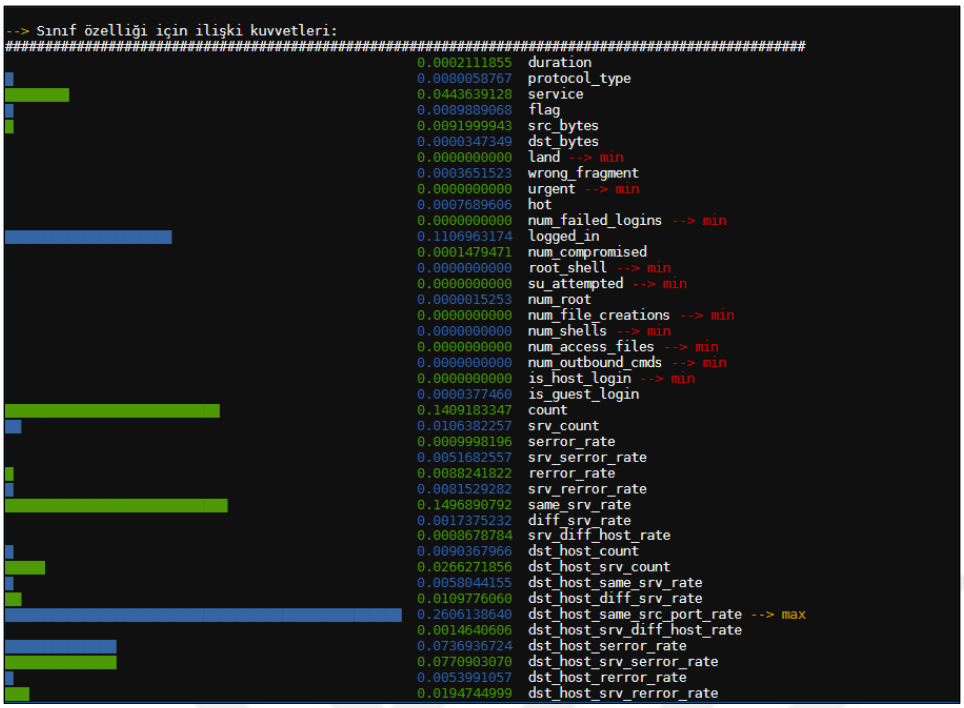
Aynı ayırma noktalarını eğitim kümesinde, test kümesinde ve trafik kayıtları üzerinde kullanabilmek için kayıt modülünden faydalanılmıştır. Eğitim kümesinin önişlem adımında her öznitelik için hesaplanan dinamik ayırma noktaları veri tabanına kaydedilir. Trafik kayıtlarının veya test kümesinin önişlem adımlarında veri tabanından getirilen ayırma noktaları kullanılarak sürekli veriler kesikli verilere dönüştürülür. Böylece eğitim kümesinin, test kümesinin ve trafik kayıtlarının aynı değerler ile gruplanması sağlanır. Aksi takdirde benzer kümelere aynı önişlem adımları uygulanmadığından kayıtlar farklılaşacak ve veri madenciliği algoritmasının başarımı düşecektir. Çalışma kapsamında geliştirilen STS'de test verisi için kullanılan ayırma noktaları ve eğitim verisi için kullanılan ayırma noktaları ayrı hesaplandığında ID3 algoritmasının doğruluğunun %99 değerinden %88 değerine düştüğü görülmüştür.

Sınıf etiketlerinin bulunabilmesi için büyük öneme sahip özniteliklerin tespit edilmesi için ilişki kuvvetleri hesaplanmıştır. Bir özniteliğin ilişki kuvveti büyüdükçe sınıf etiketini bulmak için faydası artmaktadır. Dolayısı ile büyük ilişki kuvvetine sahip özniteliklerin seçilmesi algoritmanın performansını artırmakta ve gereksiz işlem yükünden kurtarmaktadır. Buna ek olarak ilişki kuvveti sıfır çıkan öznitelikler öğrenme sırasında kullanılmayabilir. Çünkü öğrenmeye katkıları yoktur.

Tüm bunlara ek olarak, elde edilen ilişki kuvvetleri üzerinde IQR hesaplaması yapılmıştır. IQR hesaplaması ile alt sınırdan küçük olan ilişki kuvvetlerine sahip niteliklerin seçilmemesi önerilmiştir. Literatür taramasında öznitelik seçimi için oldukça iyi sonuç alındığı görüldüğünden Extra Randomized Tree algoritması kullanılmasına karar verilmiştir [199]. Hazır kütüphaneden faydalanılarak kullanılmış ilgili algoritma ile yalnızca niteliklerin ilişki kuvvetleri hesaplanmıştır. Kullanılacak olan öznitelikler, ilgili veri kümesi üzerinde öğrenme işlemi gerçekleştirecek olan seçilmiş algoritmanın eğitilmesi aşamasında kullanıcıdan alınmaktadır. Dolayısı ile öznitelik seçme işleminin dinamik olarak yapılması sağlanmıştır.

Öznitelikler ile sınıf etiketleri arasındaki ilişki kuvvetleri işlemi

Sınıf etiketlerinin bulunabilmesi için büyük öneme sahip özniteliklerin tespit edilmesi amacıyla ilişki kuvvetleri hesaplanmış ve Resim 4.18 üzerinde görüldüğü gibi grafik üzerinde kullanıcıya sunulmuştur.



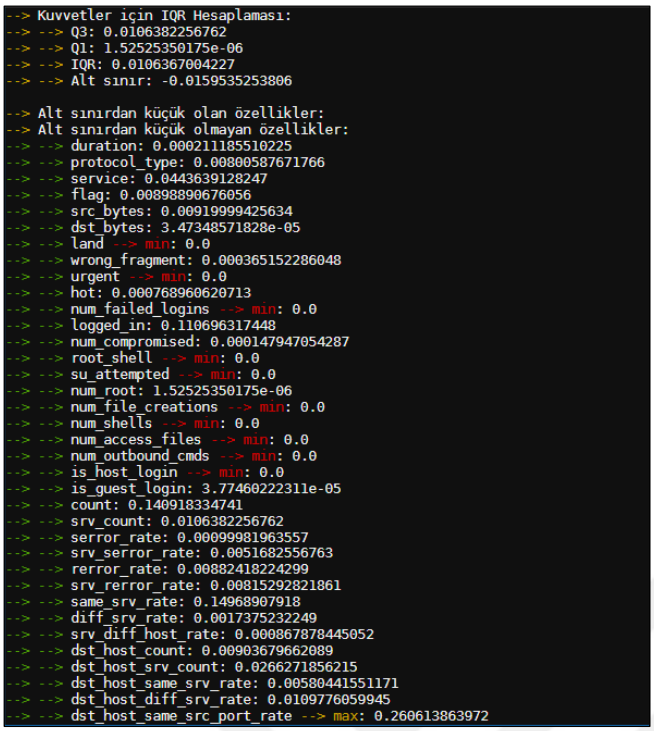
Resim 4.18 Öznitelikler ve sınıf etiketi arasındaki ilişki kuvvetleri grafiği

Tüm bunlara ek olarak, ilgili ilişki kuvvetleri üzerinde IQR hesaplaması yapılmıştır. IQR hesaplaması ile alt sınırdan küçük olan ilişki kuvvetlerine sahip özniteliklerin seçilmemesi önerilmektedir. İlgili ekran görüntüsü Resim 4.19 ile gösterilmiştir.

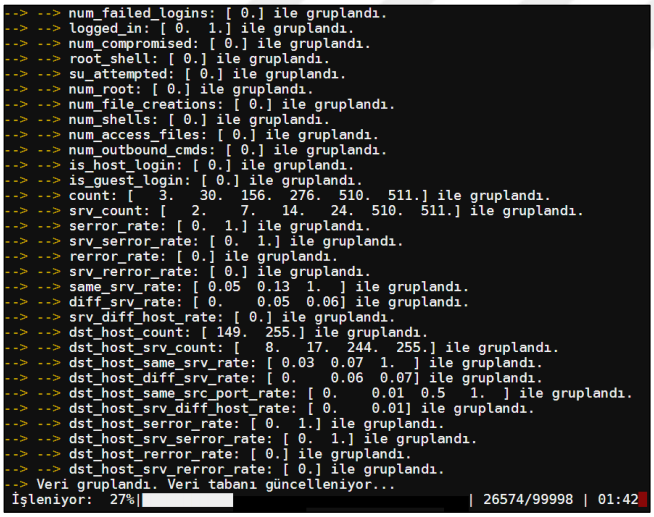
Sürekli verilerin gruplanması işlemi

Sürekli veri tipinde değerlere sahip öznitelikler listelenerek kullanıcıdan gruplandırma işlemi için onay alınır. Ardından, tüm kayıtlar veri tabanından getirilerek belleğe aktarılır. Kayıtlar üzerinde ilgili öznitelikler için ayırma noktaları hesaplanır. İlgili ayırma noktalarına göre kayıtlar veri tabanında güncellenir. Ardından, ilgili ayırma noktalarının test ve trafik kayıtları için de kullanılabilmesi için veri tabanına kaydedilir.

Hesaplanmış ayırma noktalarını gösteren ekran alıntısına Resim 4.20 üzerinde yer verilmiştir. Listelenen ayırma noktalarına göre same_srv_rate özniteliğinin 3 gruba ayrılacağı görülmektedir. İlgili öznitelik için değeri 0,05 ile eşit veya küçük olan kayıtlar birinci grup, değeri 0,13 değerine eşit veya 0,05 – 0,13 arasında olan kayıtlar ikinci grup, değeri 1 değerine eşit veya 0,13 – 1 arasında olan kayıtlar ise üçüncü grup olarak tanımlanır. Birinci grup, ikinci grup ve üçüncü grup içinde bulunan kayıtların ilgili özniteliklerinin değerleri sırası ile 0,1 ve 2 değerleri ile değiştirilir. Böylece, ayrıma noktaları belirlenir ve özniteliklerin sürekli değerleri kesikli değerlere dönüştürülür.



Resim 4.19. Öznitelikler ve sınıf etiketi arasındaki ilişki kuvvetleri dereceleri



Resim 4.20. Sürekli verilerin gruplanması adımı

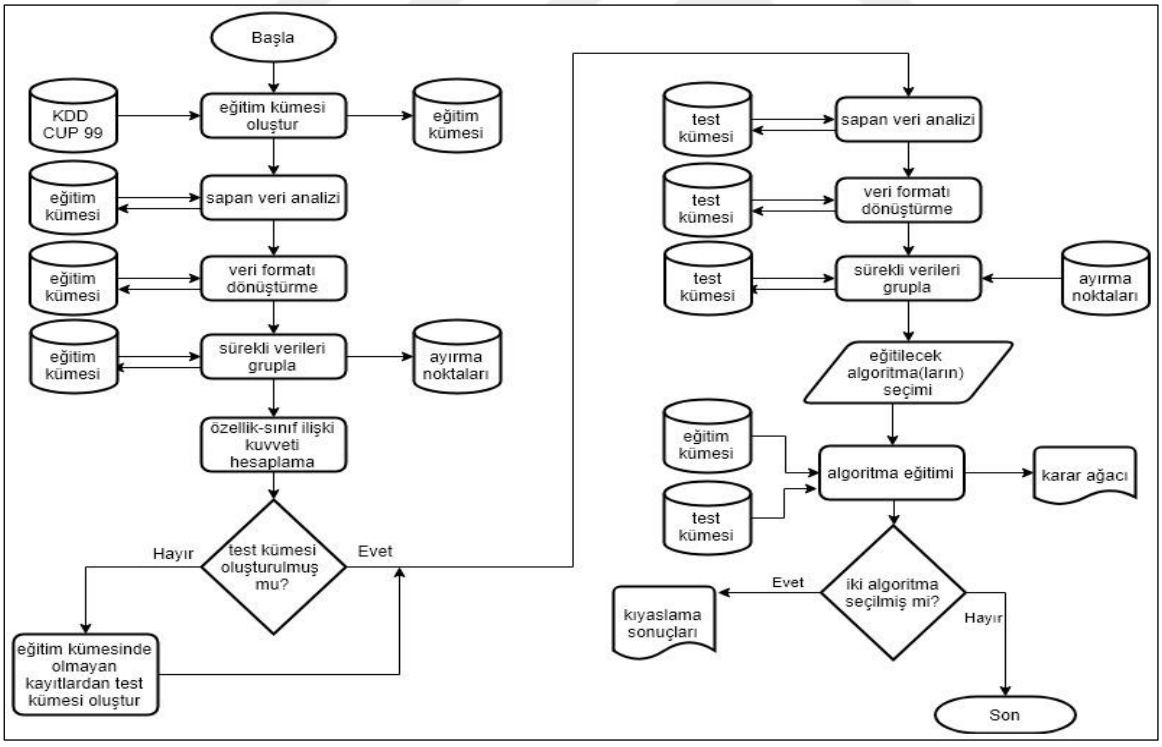
Test kümesi önişleme ve dönüştürme adımları işlemi

Test kümesi üzerinde bulunan sürekli değerler, veri tabanına kaydedilmiş ayrıma noktaları kullanılarak kesikli değerler ile değiştirilir. Buna ek olarak, test kümesinde bulunan kayıtlar için de eğitim kümesine uygulandığı gibi veri formatı dönüştürme işlemleri uygulanır. Test kümesi üzerinde gerçekleştirilen ilgili adımların ekran görüntüleri, eğitim kümesine uygulanan adımların ekran görüntüleri ile aynıdır.

4.2.4. Veri madenciliği adımı

KDD CUP 99 veri kümesi üzerinde veri seçme, önişleme ve dönüştürme adımlarının ardından CART ve ID3 karar ağacı algoritmaları kullanılarak öğrenme ve test işlemleri gerçekleştirilir. ID3 ve CART algoritmalarının her ikisinin de seçilmesi ile ilgili algoritma sonuçlarına ek olarak kıyaslama tablosu da kullanıcıya sunulur. İlgili adımlar Resim 4.21 üzerinde akış diyagramı ile gösterilmiştir.

Deneysel bir çalışma olarak, test kümesi için 99 998 adet kayıttan oluşan %2'lik örneklem ve öğrenme kümesi için 489 842 kayıt barındıran %10'luk örneklem kullanılarak algoritmalar karşılaştırılmıştır. Algoritmalar için kıyaslamalar, ilgili algoritmanın doğruluğu, öğrenme süresi, sınıflandırma süresi, yaprak sayısı ve ağaç derinliği üzerinden yapılmaktadır. Çizelge 4.5 üzerinde %10'luk ve %2'lik örneklemler ile elde edilen sonuçlar gösterilmektedir. Tüm bunlara ek olarak ilgili hesaplamalarda 41 özniteliğin tümü kullanılmıştır.



Resim 4.21. KDD CUP 99 veri kümesi üzerinde veri madenciliği akış diyagramı

Sonuçlar incelendiğinde doğruluk değerlerine göre ID3 algoritmasının daha iyi sonuç verdiğine ek olarak CART algoritmasının daha kompakt bir ağaç sunduğu ancak öğrenme ve sınıflandırma hızının daha düşük olduğu görülmüştür. ID3 algoritmasının CART

algoritmasına karşın daha iyi sonuç vermesi, kullanılan veri kümesinde bulunan kayıtların veri tipinden kaynaklanabileceği gibi kütüphaneden yardım alınmadan gerçekleştirilen önişlem adımlarının ID3 algoritması için daha uygun olabileceğinden de kaynaklanabilmektedir. Literatür taramasında farklı veri kümesi üzerinde yapılan ID3 ve CART algoritmalarının karşılaştırılması sonucunda CART algoritmasının ID3 algoritmasına nazaran her zaman daha iyi sonuç verebileceği gibi bir durumun olmadığı ve ID3 algoritmasının da daha iyi sonuç verebileceği görülmüştür [188].

Çizelge 4.5. %10'luk örneklem üzerinde sonuçların karşılaştırılması

Öznitelik	**ID3**	**CART**
Öğrenme Süresi	574,582 saniye	1093,199 saniye
Sınıflandırma Süresi	1,948 saniye	4,176 saniye
Yaprak Sayısı	428	185
Derinlik	16	18
Doğruluk	%99,73	%97,94

Kullanılan karar ağacı algoritmaları için doğruluk değerlerine ek olarak her bir sınıfa ait F1 skoru, kesinlik (precision) ve hassasiyet (recall) değerleri hesaplanmıştır. Tüm bu değerlerin hesaplanabilmesi için gerçek pozitif (GP), yanlış pozitif (YP), gerçek negatif (GN) ve yanlış negatif (YN) değerleri kullanılmaktadır. Elde edilmiş ilgili başarım ölçütlerinin değerleri Resim 4.6 ile gösterilmiştir.

Çizelge 4.6. %10'luk örneklem üzerinde başarım kıyaslaması

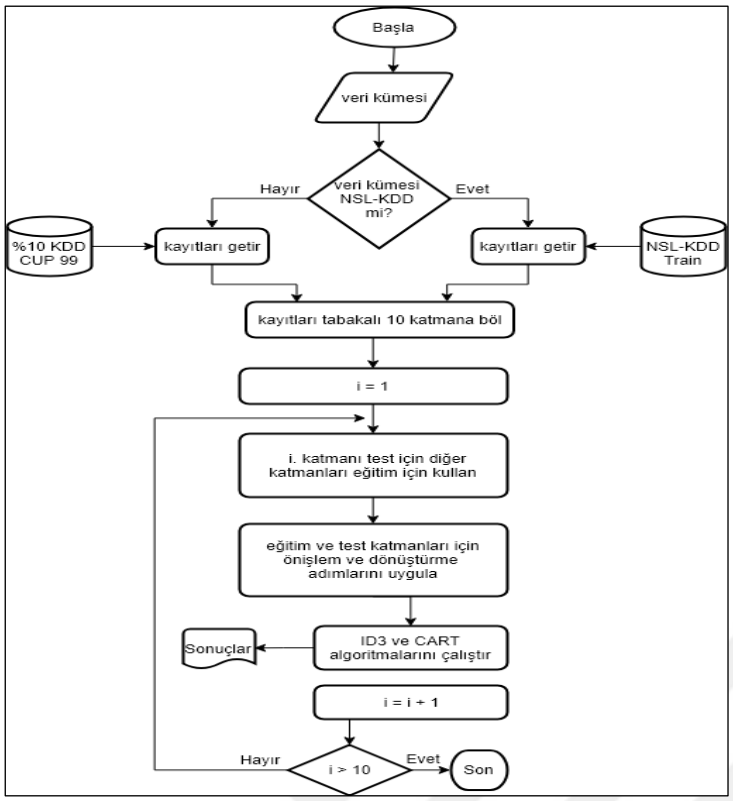
Saldırı	**ID3**			**CART**			**Sayı**
	Kesinlik	**Hassasiyet**	**F1**	**Kesinlik**	**Hassasiyet**	**F1**	
back	1,0	1,0	1,0	0,96	1,0	0,98	45
buffer_overflow	1,0	1,0	1,0	0	0	0	1
guess_passwd	1,0	1,0	1,0	1,0	1,0	1,0	1
ipsweep	0,94	0,19	0,32	0,89	0,98	0,93	255
neptune	1	1	1,0	1,0	1,0	1,0	21885
nmap	0,88	0,45	0,59	0	0	0	47
normal	0,99	1,0	0,99	1,0	0,9	0,95	19859
pod	1,0	1,0	1,0	0,83	1,0	0,91	5

Çizelge 4.6. (devam) %10'luk örneklem üzerinde başarım kıyaslaması

Saldırı	ID3			CART			Sayı
	Kesinlik	Hassasiyet	F1	Kesinlik	Hassasiyet	F1	
portsweep	0,97	0,96	0,97	0,1	0,99	0,18	213
satan	0,98	0,98	0,98	1,0	0,95	0,97	324
smurf	1,0	1,0	1,0	1,0	1,0	1,0	57322
teardrop	1,0	1,0	1,0	1,0	0,15	0,26	20
warezclient	0,89	0,81	0,85	0	0	0	21
Ortalama:	1,0	1,0	1,0	1,0	0,98	0,99	

Elde edilen sonuçlar, deneysel olarak belirlenmiş %10 değeri üzerinden hesaplandığından literatürdeki benzer çalışmalar ile karşılaştırılamaz. Literatürde, algoritmaların performansını karşılaştırabilmek amacıyla hazır %10 KDD CUP 99 veri kümesi tercih edildiği görülmüştür [177-180]. Ayrıca, ilgili veri kümesi üzerinde tabakalı 10 katlamalı çapraz doğrulama kullanıldığı bilgisi elde edilmiştir [183]. Bu kapsamda, %10 KDD CUP 99 veri kümesi kullanılarak tabakalı 10 katlamalı çapraz doğrulama hesaplanmış ve literatür ile kıyaslama yapılmıştır. Tabakalı 10 katlamalı çapraz doğrulama için akış diyagramı Resim 4.22 üzerinde gösterilmiştir.

%10 KDD CUP 99 veri kümesi üzerinde önerilen sistem ile uygulanmış benzer adımlar veri madenciliğinde güvenilirliğini kanıtlamış Weka aracı üzerinde de gerçekleştirilerek tabakalı 10 katlamalı çapraz doğrulama yapılmıştır. Tabakalı 10 katlamalı çapraz doğrulama ile kayıtlar katmanlara bölünürken rastgelelik de kullanıldığından önerilen sistem ile weka aracının ayırmış olduğu katmanlar aynı kayıtları içermemektedir. Dolayısı ile iki araç arasındaki küçük başarım farkları göz ardı edilebilir. Weka aracı ile %10 KDD CUP 99 veri kümesi üzerinde gerçekleştirilen tabakalı 10 katlamalı çapraz doğrulama işlemi sonuçları incelendiğinde ID3 algoritması için %99,9 ve CART algoritması için %99,9 ortalama doğruluk değerleri elde edilmiştir. Sonuç olarak, önerilen sistemin başarımları ile Weka aracının başarımları benzerlik göstermiştir. Bu durum, hiçbir hazır bileşen kullanılmadan gerçekleştirilmiş algoritmaların gerektiği gibi çalıştığını gözler önüne sermiştir.



Resim 4.22. Tabakalı 10 katlamalı çapraz doğrulama için akış diyagramı

Geliştirilmiş STS kullanılarak %10 KDD CUP 99 veri kümesi üzerinde gerçekleştirilen tabakalı 10 katlamalı çapraz doğrulama işlemi sonuçları incelendiğinde ID3 algoritması için %99,4 ve CART algoritması için %99,9 ortalama doğruluk değerleri elde edilmiştir. ID3 algoritması ve CART algoritması için %10 KDD CUP 99 veri kümesi üzerinde önerilen sistem ile elde edilmiş başarımlar, en az literatürdeki benzer çalışmaların başarımları kadar yüksektir [200 – 205]. Böylece, literatürdeki benzer çalışmalar ile önerilen sistemin başarımının benzerlik göstermesinden dolayı veri madenciliğinde kullanılan önişleme ve dönüştürme adımlarının başarılı bir şekilde gerçekleştirildiği görülmüştür.

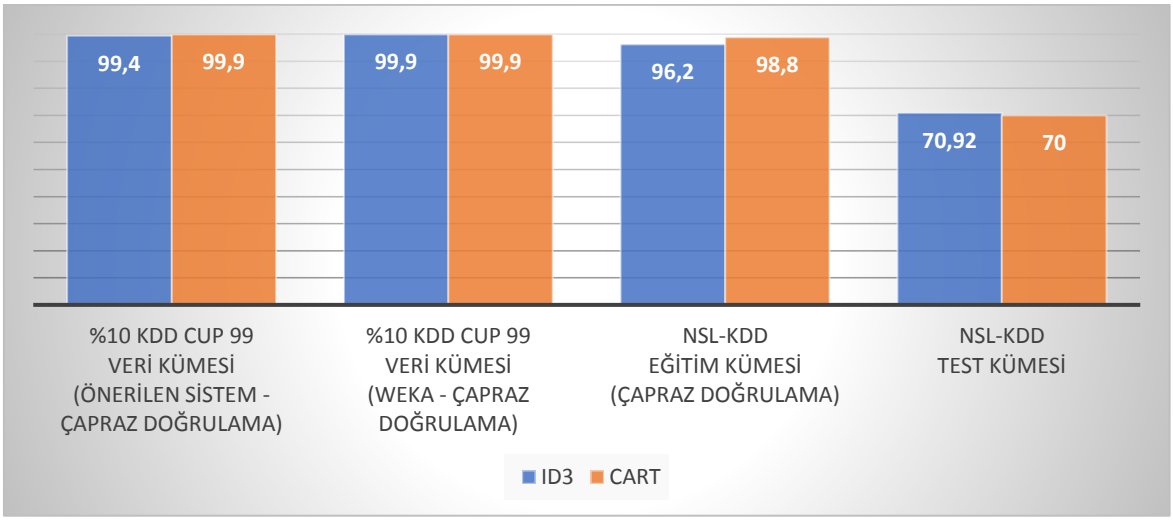
Tüm bu bilgiler ışığında önerilen sistemin, Weka aracı ve literatürdeki benzer çalışmalar ile paralellik göstermesinden dolayı veri madenciliği işlemleri için başarılı bir şekilde geliştirilmiş olduğu ispatlanmıştır.

KDD CUP 99 ve %10 KDD CUP 99 veri kümelerinin barındırdığı dezavantajlardan dolayı gerçek trafik kayıtlarının sınıflandırılmasında kullanılacak algoritmaların saldırıları öğrenmesi NSL-KDD veri kümesi üzerinden yapılmıştır. Literatür taramasında; ilgili veri kümesi üzerinde çalıştırılan algoritmaların başarımlarını karşılaştırmak için NSL-KDD

Eğitim veri kümesinin eğitim kümesi ve NSL-KDD Test veri kümesinin ise test kümesi olarak kullanıldığı veya NSL-KDD Eğitim veri kümesi üzerinde tabakalı 10 katlamalı çapraz doğrulama gerçekleştirildiği görülmüştür [188, 189, 190]. Bu kapsamda önerilen sistemin başarımı kıyaslayabilmek için her iki yöntem uygulanmıştır.

Geliştirilmiş STS ile NSL-KDD Eğitim veri kümesi üzerinde gerçekleştirilen tabakalı 10 katlamalı çapraz doğrulama işlemi sonuçları incelendiğinde ID3 algoritması için %96,2 ve CART algoritması için %98,8 ortalama doğruluk değerleri elde edilmiştir. ID3 algoritması ve CART algoritması için NSL-KDD Eğitim veri kümesi üzerinde önerilen sistem ile elde edilmiş başarımların, literatürdeki çalışmalar ile benzer doğrulukta olduğu görülmüştür [190, 200, 206].

Geliştirilmiş STS, NSL-KDD Eğitim veri kümesi üzerinden saldırı tiplerini öğrenerek NSL-KDD Test veri kümesini sınıflandırmıştır. NLS-KDD Test veri kümesinin NSL-KDD Eğitim veri kümesinde bulunmayan farklı saldırılarıda barındırması ile önerilen sistemin bilinmeyen saldırılar ile karşılaşıldığı durumlarda başarımının ölçülmesi sağlanmıştır. Elde edilen sonuçlar incelendiğinde CART algoritmasının %70, ID3 algoritmasının ise %70,92 doğruluk ile çalıştığı gözlenmiştir. Yapılan literatür taramasında, önerilen sistemin başarımının en az benzer çalışmalarda elde edilmiş başarımlar kadar yüksek olduğu görülmüştür. [207, 208] Elde edilmiş tüm başarımlar Resim 4.23 üzerinde gösterilmiştir.



Resim 4.23. Algoritmaların veri kümelerine ve araçlara göre başarımları

NSL-KDD Eğitim veri kümesi, gerçek trafik kayıtlarını sınıflandırmak için saldırı tiplerini öğrenme amacıyla tercih edildiğinden güncelliğinin test edilmesine ihtiyacı duyulmuştur. Çünkü elde edilen sınıflandırma sonuçları değerlendirilerek STİ üretilmektedir. NSL-KDD

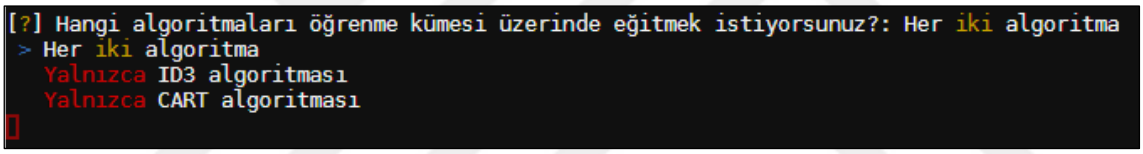
Eğitim veri kümesinde bulunan saldırı tipleri, KDD CUP 99 veri kümesinde bulunan saldırı tipleri ile aynıdır. İlgili veri kümelerinin sahip olduğu saldırı tipleri Çizelge 3.1 üzerinde gösterilmiştir. Bu kapsamda, NSL-KDD Eğitim veri kümesinde bulunan saldırı tiplerine neptune, guess_passwd, buffer_overflow, ipsweep ve portsweep örnek olarak gösterilebilir.

Rusma Mulyadi tarafından gerçekleştirilen bir çalışmada bahsi geçen saldırı tipleri açıklanmıştır [209]. İlgili çalışmada, neptune saldırısının bir veya daha fazla port kullanılarak gerçekleştirilen Syn-Flood servis dışı bırakma saldırısı olduğu ifade edilmektedir. Buna ek olarak guess_passwd saldırısının, telnet üzerinden misafir kullanıcısının parolasını tahmin etme işlemi olarak tanımlandığı görülmüştür. buffer_overflow saldırısının ise UNIX işletim sistemleri üzerinde gerçekleştirilen arabellek aşımı saldırıları olduğuna değinilmiştir. Tüm bunlara ek olarak portsweep saldırısının, tek bir sistem üzerinde bulunan portların taranarak açık servislerin bulunması olarak ifade edildiği ve ipsweep saldırısının ise birden fazla adres üzerinde gerçekleştirilen portsweep saldırısı veya ping gönderme işlemi olduğuna değinildiği görülmüştür.

Bahsi geçen saldırı tipleri ve gerçekleştirme adımları incelendiğinde saldırıların günümüzde de aynı yöntem ve adımlar üzerinden gerçekleştirildiği görülmektedir. Buna rağmen, yıllar içinde bazı saldırıların gerçekleştirilmesinde kullanılan yöntemlerde değişiklikler gözlenebilir. Bu sebeple, NSL-KDD veri kümesinde bulunan saldırıların güncelliğinin test edilmesine ihtiyaç duyulmuştur. Ayrıca NSL-KDD Eğitim veri kümesi, gerçek trafik kayıtlarını sınıflandırmak için saldırı tiplerini öğrenme amacıyla tercih edildiğinden ve ilgili sınıflandırma sonuçları doğrultusunda STİ üretildiğinden saldırıların güncelliğinin test edilmesi önem arz etmektedir. Ancak, saldırıların gerçekleştirilmesi ve ele alınan saldırı ile ilgilisiz olan gerçek trafik kayıtların tespit edilerek veri kümesinin temizlenmesi için uzman bilgisine ihitiyaç duyulmaktadır. Bu durum, NSL-KDD Eğitim veri kümesinde bulunan tüm saldırı tiplerinin güncel olduğunun ispatlanmasını oldukça zahmetli ve kapsamlı bir çalışma haline getirmektedir. Tüm bunlara ek olarak, bazı saldırı tiplerini gerçekleştirmek için büyük bir ağa ihtiyaç duyulmaktadır. Dolayısı ile bu çalışma kapsamında tüm saldırılar ele alınamamıştır. Çalışmada araştırmacının uzmanlığı doğrultusunda ilgili test işlemi için portsweep saldırısı tercih edilmiştir. Yapılan analizler neticesinde %99,50 doğrulukta sonuç elde edilerek portsweep saldırısının başarılı bir şekilde tespit edilebildiği ve NLS-KDD Eğitim veri kümesinin ilgili saldırı için güncel olduğu ispatlanmıştır.

Algoritmaların eğitilmesi işlemi

Veri madenciliği algoritmaları tarafından kullanılacak olan özniteliklerin seçilmesi işlevi algoritmaların eğitiminde de sunulmuştur. Bu sayede sınıf özniteliği ile yüksek ilişki kuvvetine sahip özniteliklerin seçilmesine olanak sağlanmıştır. Kullanılacak özniteliklerin seçimi adımının ardından, eğitim kümesi üzerinde çalışarak ilgili sınıf etiketlerini öğrenecek olan karar ağacı algoritması belirlenir. Tek bir karar ağacı algoritmasının seçilmesi durumunda, ilgili algoritma ile eğitim kümesi kullanılarak saldırı tipleri öğrenilir, test kümesi üzerinden başarımı hesaplanır ve üretilmiş karar ağacına ait öğrenme süresi, sınıflandırma süresi ve yaprak sayısı gibi veriler ekrana yansıtılır. Her iki algoritmanın da eğitilmesi tercih edildiği durumlarda ise her ağaç verisine ek olarak karşılaştırma tablosu ekranda gösterilir. Resim 4.24 üzerinde her iki algoritmanın seçilme aşaması görülmektedir. Tüm bu işlemlerin ardından kullanıcıya elde edilmiş karar ağacının görselleştirilmesi için seçenek sunulur.



Resim 4.24. Öğrenme adımı için kullanılacak algoritmanın belirlenmesi

İlgili ekran görüntülerinin sunulması amacıyla deneysel bir çalışma olarak, test kümesi için 99 998 adet kayıttan oluşan %2'lik örneklem ve eğitim kümesi için 99 998 kayıt barındıran %2'lik başka bir örneklem kullanılarak algoritmalar karşılaştırılmıştır. Çizelge 4.7 üzerinde %2'lik örneklemler üzerinde 41 özniteliğin tümünün kullanılması ile elde edilen sonuçlar gösterilmektedir.

Çizelge 4.7. %2'lik örneklem üzerinde sonuçların karşılaştırılması

Öznitelik	**ID3**	**CART**
Öğrenme Süresi	76,539 saniye	138,479 saniye
Sınıflandırma Süresi	1,639 saniye	2,927 saniye
Yaprak Sayısı	187	89
Derinlik	9	15
Doğruluk	%99,74	%82,97

İlgili sonuçlar ile %10'luk ve %2'lik örneklem ile elden edilen sonuçlar karşılaştırıldığında, eğitim kümesinin boyutunun artırılmasına rağmen ID3 algoritmasının başarımının aynı kaldığı fakat CART algoritmasının başarımının oldukça arttığı görülmüştür. Bu durum %2'lik örneklem içinde bulunan kayıtların %10'luk örneklemde bulunan kayıtlara nazaran saldırıları yeterli derecede ifade edemediğinden kaynaklanabilirdir. Tüm bunlara ek olarak, kullanılan karar ağacı algoritmaları için her bir sınıfa ait F1 skoru, kesinlik ve hassasiyet değerleri hesaplanmaktadır. %2'lik örneklem üzerinde öğrenme neticesinde elde edilmiş ilgili başarım ölçütlerinin değerleri Çizelge 4.8 ile gösterilmiştir.

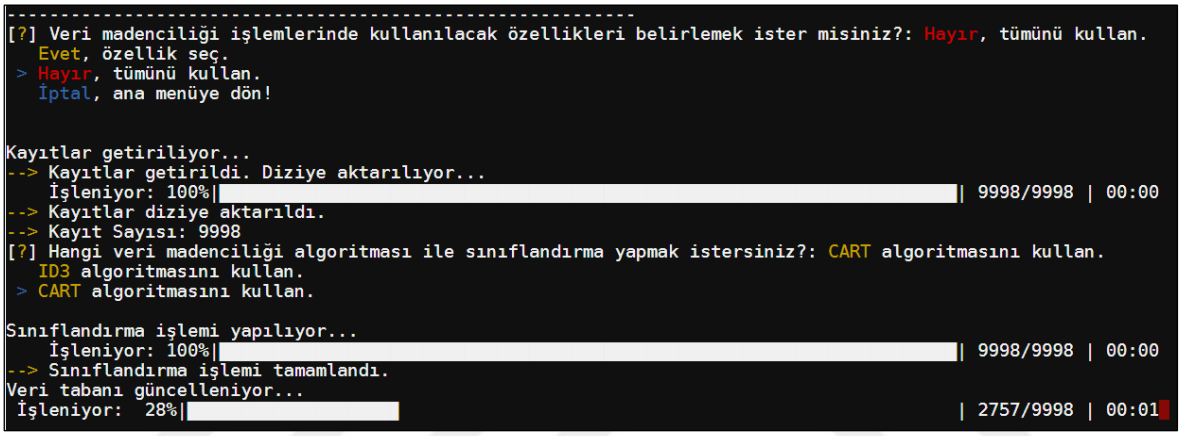
Çizelge 4.8 %2'lik örneklem üzerinde başarım kıyaslaması

Saldırı	**ID3**			**CART**			**Sayı**
	kesinlik	**hassasiyet**	**F1**	**kesinlik**	**hassasiyet**	**F1**	
back	1,0	1,0	1,0	0,03	1,0	0,06	45
buffer_overflow	0	0	0	0	0	0	1
guess_passwd	1,0	1,0	1,0	0	0	0	1
ipsweep	0,88	0,29	0,44	0,88	0,98	0,93	255
neptune	1,0	1,0	1,0	1,0	1,0	1,0	21885
nmap	1,0	0,43	0,60	1,0	0,02	0,04	47
normal	0,99	1,0	0,99	0,99	0,15	0,26	19859
pod	1,0	1,0	1,0	1,0	1,0	1,0	5
portsweep	0,98	0,96	0,97	0,01	1,0	0,03	213
satan	0,99	0,95	0,97	0,97	0,94	0,96	324
smurf	1,0	1,0	1,0	1,0	1,0	1,0	57322
teardrop	1,0	1,0	1,0	1,0	0,1	0,18	20
warezclient	0,79	0,71	0,75	0	0	0	21
Ortalama:	1,0	1,0	1,0	1,0	0,83	0,85	

Test kümesi sınıflandırma işlemi

Oluşturulmuş test kümesi, algoritmaların başarımlarını hesaplamak için kullanılmaktadır. Algoritmalar tarafından doğru sınıf etiketi ile tahmin edilmiş kayıt sayısı üzerinden başarım sonuçları elde edilir. Ancak, başarımların hesaplanmasında tüm sınıflandırma sonuçları bellek üzerinde tutulup veri tabanı üzerinde bulunan kayıtlar

güncellenmemektedir. İlgili adım ile test kümesindeki kayıtlar için tahmin edilmiş sınıf etiketleri veri tabanı üzerinde güncellenir. Böylece, kullanılan algoritmanın ilgili kayıtlar için hangi sınıflandırmayı yaptığı incelenebilir. Dolayısı ile algoritmaların sınıflandırma performanslarını kontrol etmek için kullanılabilir. Test kayıtlarının sınıflandırılmasında ilgili tablonun prediction sütunu güncellenmektedir. Sınıf özniteliği ve diğer öznitelikler üzerinde bir değişiklik yapılmadığından aynı test verisinin bozulmadan tekrar kullanılmasına imkân sağlanmıştır. Resim 4.25 üzerinde sınıflandırma işlemi ekran görüntüsü sunulmuştur.



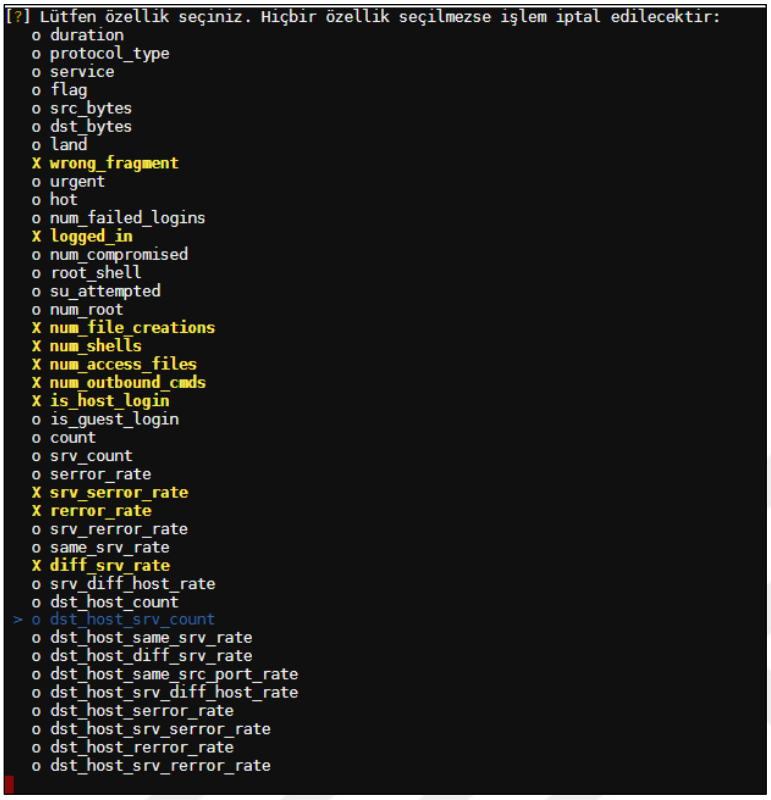
Resim 4.25. Test kümesi sınıflandırma adımı

Kullanıcının sınıflandırma algoritmalarının kullanmasını istediği öznitelikleri seçmesi ile veri tabanından ilgili kayıtlar belleğe aktarılır. Belleğe aktarılan kayıtlar üzerinde çalıştırılacak karar ağacı algoritmasının belirtilmesi ile ilgili algoritmaya ait olan kaydedilmiş ağaç dosyadan okunarak sınıflandırma işlemi yapılır. Veri madenciliği algoritmalarının kullanacağı özniteliklerin seçilmesi Resim 4.26 üzerinde gösterilmiştir. İlgili ekran görüntüsü ile görüldüğü üzere seçim kutuları ile birden çok öznitelik seçilmesine imkan sağlanmaktadır.

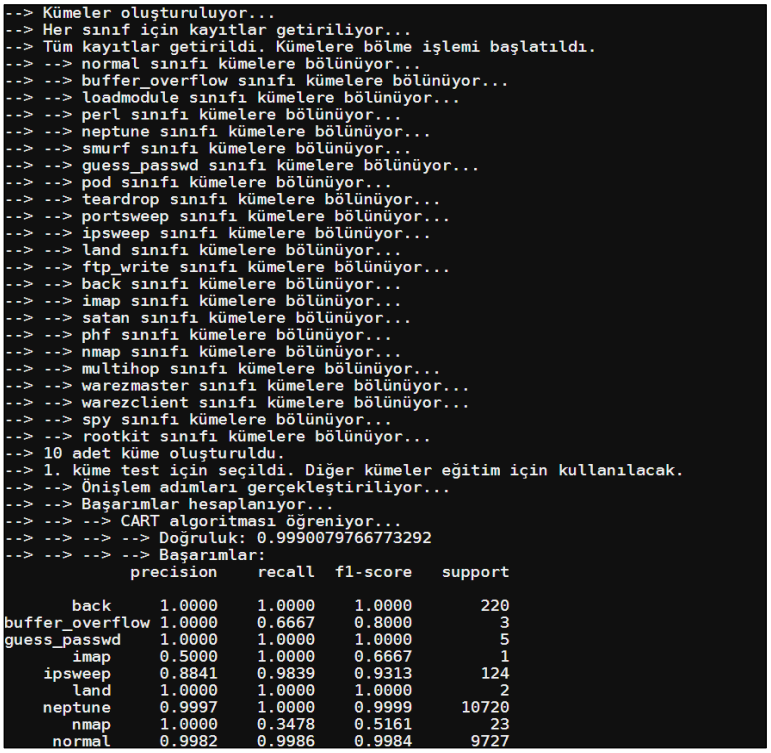
KDD CUP 99 %10 veri kümesi üzerinde çapraz doğrulama işlemi

KDD CUP 99 %10 veri kümesi veri tabanında bulunan ilgili tablo üzerinden getirilerek bellek üzerine aktarılır. Bellek üzerine aktarılan kayıtlar, katmanlı örneklem çıkarma işlemi ile 10 adet kümeye bölünür. Oluşturulmuş kümelerden biri test ve diğerleri eğitim kümesi olarak kullanılmak üzere sıra ile bir döngü içinde 10 kez başarımlar hesaplanır. Her adımda, başarımlara ek olarak doğruluk matrisi de ekranda gösterilir. Tüm bu işlemlerin ardından, hesaplanmış başarımların aritmetik ortalaması alınarak algoritmalar için ortalama

doğruluk değerleri kullanıcıya sunulur. Resim 4.27 üzerinde ilgili adım için ekran görüntüsü sunulmuştur.



Resim 4.26. Özniteliklerin seçim kutuları ile belirlenmesi adımı



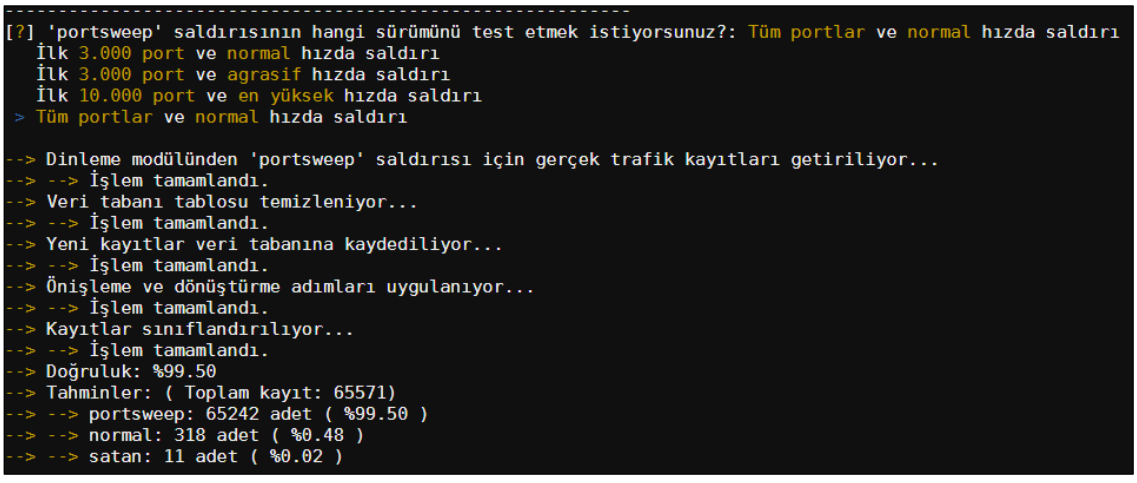
Resim 4.27. KDD CUP 99 %10 veri kümesi çapraz doğrulama sonuçları

NSL-KDD Eğitim veri kümesi saldırı tiplerinin güncelliğinin test edilmesi işlemi

portsweep saldırısı için test kayıtlarının elde edilmesinde tehdit simülasyon modülü ve zafiyet simülasyon modülü kullanılmıştır. portsweep saldırısını gerçekleştirmek için tehdit simülasyon modülü üzerinde nmap aracından faydalanılarak "nmap zafiyet_simülasyon_modülü_ağı -p-" komutu çalıştırılmıştır. Böylece zafiyet simülasyon modülünün tüm portları taranarak portsweep saldırısı gerçekleştirilmiştir.

Tüm bunlara ek olarak, deneysel olarak belirlenmiş ilk 3 000 ve 10 000 port üzerinden nmap aracı ile farklı saldırı hızlarında portsweep saldırıları gerçekleştirilmiş ve kaydedilmiştir. Kullanıcı tarafından seçilen gerçek kayıtlar, ağ trafiği dinleme modülü üzerinden istenilen formatta elde edilir. Veri tabanında bulunan ilgili tablo temizlenir ve elde edilmiş kayılar tabloya eklenir. Tabloda bulunan kayıtlar üzerinde önişleme ve dönüştürme adımları uygulanarak kayıtlar sınıflandırılır. Sınıflandırma işlemi neticesinde doğru saldırı tipi ile etiketlenen kayıt sayısı üzerinden başarım hesaplanır. Ayrıca, başka saldırı tipi olarak etiketlenmiş kayıtların sınıfları ve sayıları da listelenir.

Resim 4.28 üzerinde tehdit simülasyon modülü ile zafiyet simülasyon modülüne tüm portları kullanarak gerçekleştirilen portsweep saldırısı için başarımların hesaplandığı ekran görüntüsü gösterilmiştir.



Resim 4.28. NSL-KDD Eğitim kümesi saldırı tipleri güncellik testi adımı

İlgili ekran görüntüsü üzerinde elde edilmiş %99,50 doğruluktaki sonuç ile portsweep saldırısının başarılı bir şekilde tespit edilebildiği ve NLS-KDD Eğitim veri kümesinin ilgili saldırı için güncel olduğu ispatlanmıştır.. Nmap aracının sağladığı saldırı hızlarının da kullanılması ile gerçekleştirilen saldırı sonuçları Çizelge 4.9 üzerinde gösterilmiştir.

İlk 3 000 port üzerinden gerçekleştirilen normal hızdaki portsweep saldırısı için %89,29 doğruluk elde edilmiştir. Buna karşın, aynı portlar kullanılarak gerçekleştirilen agresif hızdaki portsweep saldırısı için %89,53 doğruluk elde edilmiştir. Böylece, saldırı hızının artması ile STS'nin başarımının arttığı görülmüştür. Ayrıca, ilk 10 000 port üzerinden gerçekleştirilen en yüksek hızdaki saldırı için ise doğruluğun %96,78'e ulaştığı tespit edilmiştir. Buna ek olarak, normal hızdaki 65 535 port üzerinden gerçekleştirilen saldırı neticesinde %99,50 doğruluk elde edilmiştir. Bu durum, saldırı trafiğinin boyutunun artması ile STS'nin başarımının arttığı olarak ifade edilebilir. Dolayısıyla bu çalışma ile STS'nin başarımının, incelenen saldırı trafiğinin boyutu ve gerçekleşme hızı ile doğru orantılı olduğu sonucuna da ulaşılmıştır.

Çizelge 4.9. Farklı port sayısı ve saldırı hızı ile portsweep başarımları

Nmap Hızı	**Port Sayısı**	**Sonuçlar**		
		Etiket	**Başarım**	**Sayı**
Normal (T3)	3 000	portsweep	%89,29	2701
		normal	%10,45	316
		satan	%0,26	8
		Toplam kayıt sayısı: 3025		
Agresif (T4)	3 000	portsweep	%89,53	2669
		normal	%10,37	309
		satan	%0,07	2
		neptune	%0,03	1
		Toplam kayıt sayısı: 2981		
Çılgın (T5)	10 000	portsweep	%96,78	9491
		normal	%3,10	304
		satan	%0.12	12
		Toplam kayıt sayısı: 9807		
Normal (T3)	65 535	portsweep	%99,50	65242
		normal	%0,48	318
		satan	%0,02	11
		Toplam kayıt sayısı: 65571		

NSL-KDD Eğitim veri kümesi üzerinde çapraz doğrulama işlemi

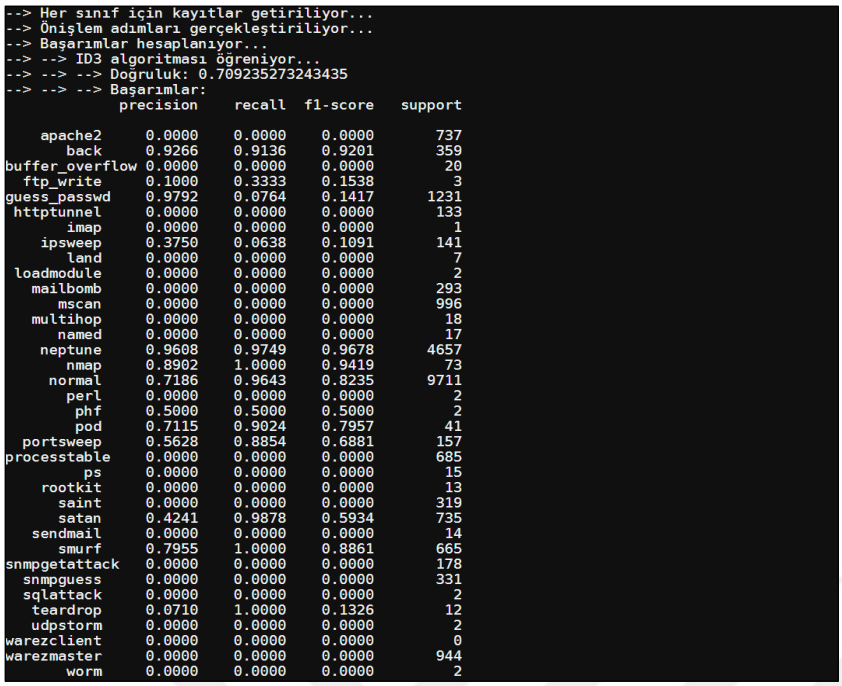
NSL-KDD Eğitim veri kümesi veri tabanında bulunan ilgili tablo üzerinden getirilerek bellek üzerine aktarılır. KDD CUP 99 %10 veri kümesi kayıtlarına uygulandığı gibi ilgili kayıtlar katmanlı örneklem çıkarma işlemi ile 10 adet kümeye bölünerek kümelerden biri test ve diğerleri eğitim kümesi olarak kullanılmak üzere sıra ile bir döngü içinde 10 kez başarımlar ve doğruluk matrisi hesaplanır. Yapılan tekrarlar üzerinden elde edilmiş başarımların aritmetik ortalaması hesaplanarak algoritmalar için ortalama doğruluk değerleri kullanıcıya sunulur. Resim 4.29 üzerinde ilgili adım için ekran görüntüsü sunulmuştur.

```
--> --> warezclient sınıfı kümelere bölünüyor...
--> --> spy sınıfı kümelere bölünüyor...
--> --> rootkit sınıfı kümelere bölünüyor...
--> 10 adet küme oluşturuldu.
--> 1. küme test için seçildi. Diğer kümeler eğitim için kullanılacak.
--> --> Önişlem adımları gerçekleştiriliyor...
--> --> Başarımlar hesaplanıyor...
--> --> --> CART algoritması öğreniyor...
--> --> --> --> Doğruluk: 0.9860195408690127
--> --> --> --> Başarımlar:
              precision    recall  f1-score   support

        back     0.9691    0.9895    0.9792        95
buffer_overflow 1.0000    0.6667    0.8000         3
  ftp_write     0.0000    0.0000    0.0000         0
guess_passwd    1.0000    1.0000    1.0000         5
        imap    1.0000    1.0000    1.0000         1
     ipsweep    0.7980    0.8802    0.8371       359
        land    0.5000    1.0000    0.6667         1
     neptune    0.9993    1.0000    0.9996      4121
        nmap    0.5794    0.4899    0.5309       149
      normal    0.9969    0.9948    0.9958      6734
         pod    1.0000    0.9500    0.9744        20
  portsweep     0.9932    0.9932    0.9932       293
     rootkit    0.0000    0.0000    0.0000         1
       satan    0.9861    0.9807    0.9834       363
       smurf    0.9962    0.9962    0.9962       264
    teardrop    1.0000    1.0000    1.0000        89
warezclient     0.9529    0.9101    0.9310        89
warezmaster     1.0000    1.0000    1.0000         2

--> --> --> ID3 algoritması öğreniyor...
--> --> --> --> Doğruluk:0.9607593931209786
--> --> --> --> Başarımlar:
              precision    recall  f1-score   support

        back     1.0000    1.0000    1.0000        95
buffer_overflow 1.0000    0.6667    0.8000         3
guess_passwd    1.0000    1.0000    1.0000         5
        imap    1.0000    1.0000    1.0000         1
     ipsweep    0.9355    0.1616    0.2755       359
        land    0.0000    0.0000    0.0000         1
     neptune    0.9998    0.9995    0.9996      4121
        nmap    0.9792    0.3154    0.4772       149
```

Resim 4.29. NSL-KDD Eğitim veri kümesi çapraz doğrulama sonuçları

NSL-KDD veri kümesi üzerinde başarımların hesaplanması

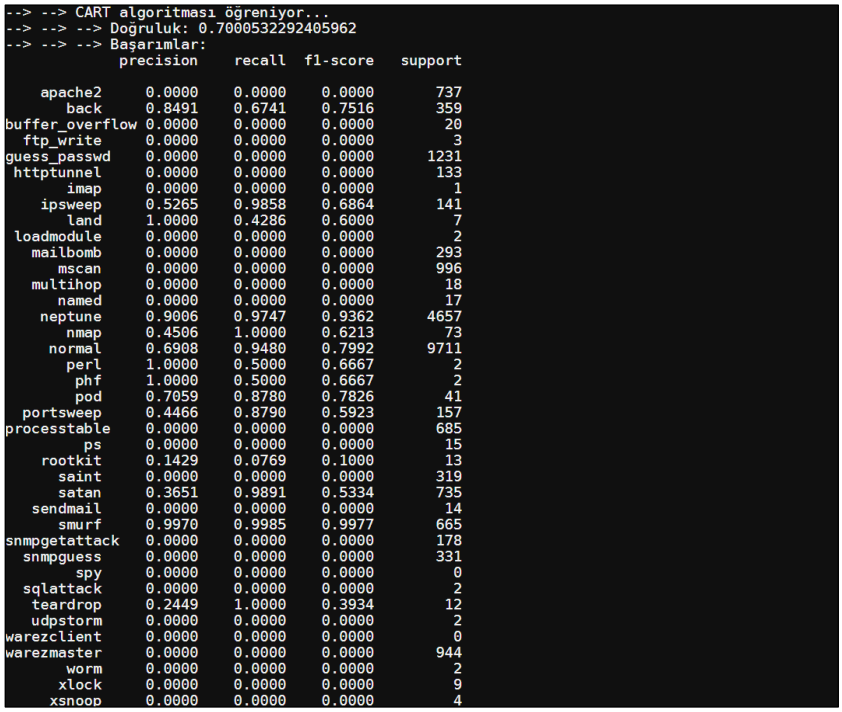
NSL-KDD Eğitim veri kümesi eğitim kümesi olarak ve NSL-KDD Test veri kümesi ise test kümesi olarak kullanılarak veri madenciliği algoritmalarının başarımları hesaplanır. Resim 4.30 ve Resim 4.31 üzerinde ilgili algoritmaların başarımlarını gösteren ekran görüntüleri sunulmuştur.



```
--> Her sınıf için kayıtlar getiriliyor...
--> Önişlem adımları gerçekleştiriliyor...
--> Başarımlar hesaplanıyor...
--> --> ID3 algoritması öğreniyor...
--> --> --> Doğruluk: 0.709235273243435
--> --> --> Başarımlar:
              precision    recall  f1-score   support

      apache2     0.0000    0.0000    0.0000       737
         back     0.9266    0.9136    0.9201       359
buffer_overflow 0.0000    0.0000    0.0000        20
    ftp_write     0.1000    0.3333    0.1538         3
guess_passwd     0.9792    0.0764    0.1417      1231
  httptunnel     0.0000    0.0000    0.0000       133
        imap     0.0000    0.0000    0.0000         1
     ipsweep     0.3750    0.0638    0.1091       141
        land     0.0000    0.0000    0.0000         7
  loadmodule     0.0000    0.0000    0.0000         2
    mailbomb     0.0000    0.0000    0.0000       293
       mscan     0.0000    0.0000    0.0000       996
    multihop     0.0000    0.0000    0.0000        18
       named     0.0000    0.0000    0.0000        17
     neptune     0.9608    0.9749    0.9678      4657
        nmap     0.8902    1.0000    0.9419        73
      normal     0.7186    0.9643    0.8235      9711
        perl     0.0000    0.0000    0.0000         2
         phf     0.5000    0.5000    0.5000         2
         pod     0.7115    0.9024    0.7957        41
   portsweep     0.5628    0.8854    0.6881       157
processstable    0.0000    0.0000    0.0000       685
          ps     0.0000    0.0000    0.0000        15
     rootkit     0.0000    0.0000    0.0000        13
       saint     0.0000    0.0000    0.0000       319
       satan     0.4241    0.9878    0.5934       735
    sendmail     0.0000    0.0000    0.0000        14
       smurf     0.7955    1.0000    0.8861       665
snmpgetattack    0.0000    0.0000    0.0000       178
   snmpguess     0.0000    0.0000    0.0000       331
   sqlattack     0.0000    0.0000    0.0000         2
    teardrop     0.0710    1.0000    0.1326        12
    udpstorm     0.0000    0.0000    0.0000         2
warezclient      0.0000    0.0000    0.0000         0
warezmaster      0.0000    0.0000    0.0000       944
        worm     0.0000    0.0000    0.0000         2
```

Resim 4.30. NSL-KDD veri kümesi ID3 algoritması başarımları

İlgili veri tabanı tablolarından getirilen kayıtlar bellek üzerine aktarılır. İlgili kayıtlar için önişlem ve dönüştürme adımları bellek üzerinde gerçekleştirilir. Çalıştırılan algoritmalar için kalıcı bir öğrenme söz konusu değildir. Yalnızca algoritma başarımlarını hesaplamak ve ilgili algoritmaları kıyaslamak amacı ile kullanılmıştır.



```
--> --> CART algoritması öğreniyor...
--> --> --> Doğruluk: 0.7000532292405962
--> --> --> Başarımlar:
              precision    recall  f1-score   support

      apache2     0.0000    0.0000    0.0000       737
         back     0.8491    0.6741    0.7516       359
buffer_overflow 0.0000    0.0000    0.0000        20
    ftp_write     0.0000    0.0000    0.0000         3
guess_passwd     0.0000    0.0000    0.0000      1231
  httptunnel     0.0000    0.0000    0.0000       133
        imap     0.0000    0.0000    0.0000         1
     ipsweep     0.5265    0.9858    0.6864       141
        land     1.0000    0.4286    0.6000         7
  loadmodule     0.0000    0.0000    0.0000         2
    mailbomb     0.0000    0.0000    0.0000       293
       mscan     0.0000    0.0000    0.0000       996
    multihop     0.0000    0.0000    0.0000        18
       named     0.0000    0.0000    0.0000        17
     neptune     0.9006    0.9747    0.9362      4657
        nmap     0.4506    1.0000    0.6213        73
      normal     0.6908    0.9480    0.7992      9711
        perl     1.0000    0.5000    0.6667         2
         phf     1.0000    0.5000    0.6667         2
         pod     0.7059    0.8780    0.7826        41
   portsweep     0.4466    0.8790    0.5923       157
processstable    0.0000    0.0000    0.0000       685
          ps     0.0000    0.0000    0.0000        15
     rootkit     0.1429    0.0769    0.1000        13
       saint     0.0000    0.0000    0.0000       319
       satan     0.3651    0.9891    0.5334       735
    sendmail     0.0000    0.0000    0.0000        14
       smurf     0.9970    0.9985    0.9977       665
snmpgetattack    0.0000    0.0000    0.0000       178
   snmpguess     0.0000    0.0000    0.0000       331
         spy     0.0000    0.0000    0.0000         0
   sqlattack     0.0000    0.0000    0.0000         2
    teardrop     0.2449    1.0000    0.3934        12
    udpstorm     0.0000    0.0000    0.0000         2
warezclient      0.0000    0.0000    0.0000         0
warezmaster      0.0000    0.0000    0.0000       944
        worm     0.0000    0.0000    0.0000         2
       xlock     0.0000    0.0000    0.0000         9
      xsnoop     0.0000    0.0000    0.0000         4
```

Resim 4.31. NSL-KDD veri kümesi CART algoritması başarımları

NSL-KDD Eğitim veri kümesi üzerinde algoritmaların eğitilmesi işlemi

NSL-KDD Eğitim veri kümesinin tamamı üzerinde istenilen algoritmaların seçilen öznitelikler ile öğrenme işlemi yapılmaktadır. İlgili veri tabanı tablosundan getirilen kayıtlar bellek üzerine aktarılarak önişlem ve dönüştürme adımları gerçekleştirilir. Çalıştırılan algoritmalar ile elde edilen karar ağaçları işletim sistemi üzerinde dosyalara kaydedilerek kalıcı bir öğrenme sağlanır. Böylece, kaydedilmiş ağaçlar test kümesini veya trafik kayıtlarını sınıflandırmak için kullanılabilir. Çizelge 4.10 üzerinde NSL-KDD Eğitim kümesi üzerinde algoritmaların eğitilmesi sonucunda NSL-KDD Test veri kümesi üzerinden elde edilmiş kıyaslama tablosu gösterilmektedir.

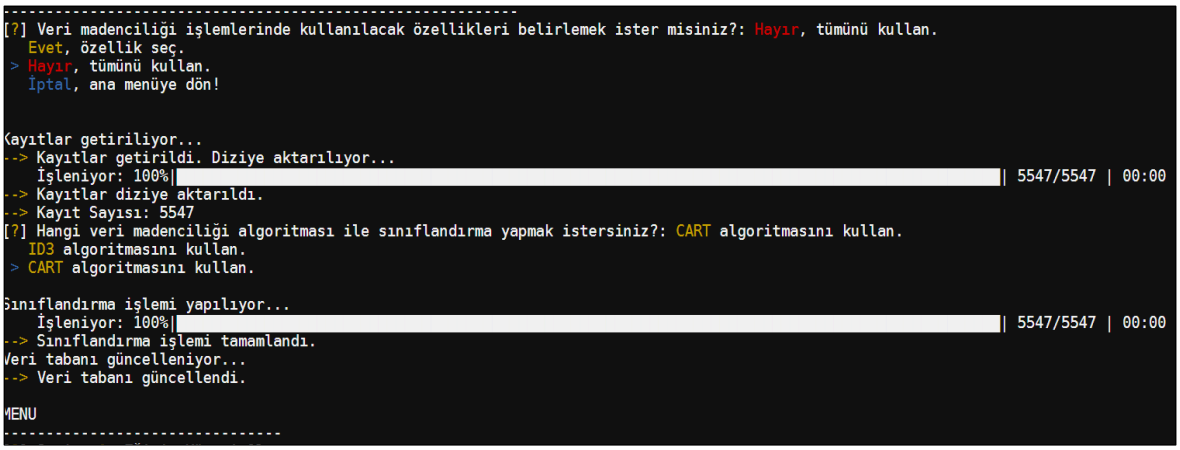
Çizelge 4.10. NSL-KDD veri kümesi algoritmaların kıyaslanması

Öznitelik	**ID3**	**CART**
Öğrenme Süresi	107,539 saniye	852,845 saniye
Sınıflandırma Süresi	0,424 saniye	1,834 saniye
Yaprak Sayısı	1206	467
Derinlik	15	23
Doğruluk	%70,92	%70,01

Mevcut trafik kayıtlarının sınıflandırılması işlemi

Mevcut trafik kayıtlarının sınıflandırılması için ilgili tabloda bulunan henüz sınıflandırılmamış veya bilinmeyen saldırı olarak sınıflandırılmış tüm kayıtlara test kümesine uygulandığı gibi önişleme ve dönüştürme adımları uygulanır. İlgili işlemin ardından, kayıtlar için seçilmiş olan karar ağacı algoritması ile sınıf etiketleri tahmin edilmekte ve ilgili kayıtlar veri tabanı tablosu üzerinde güncellenmektedir.

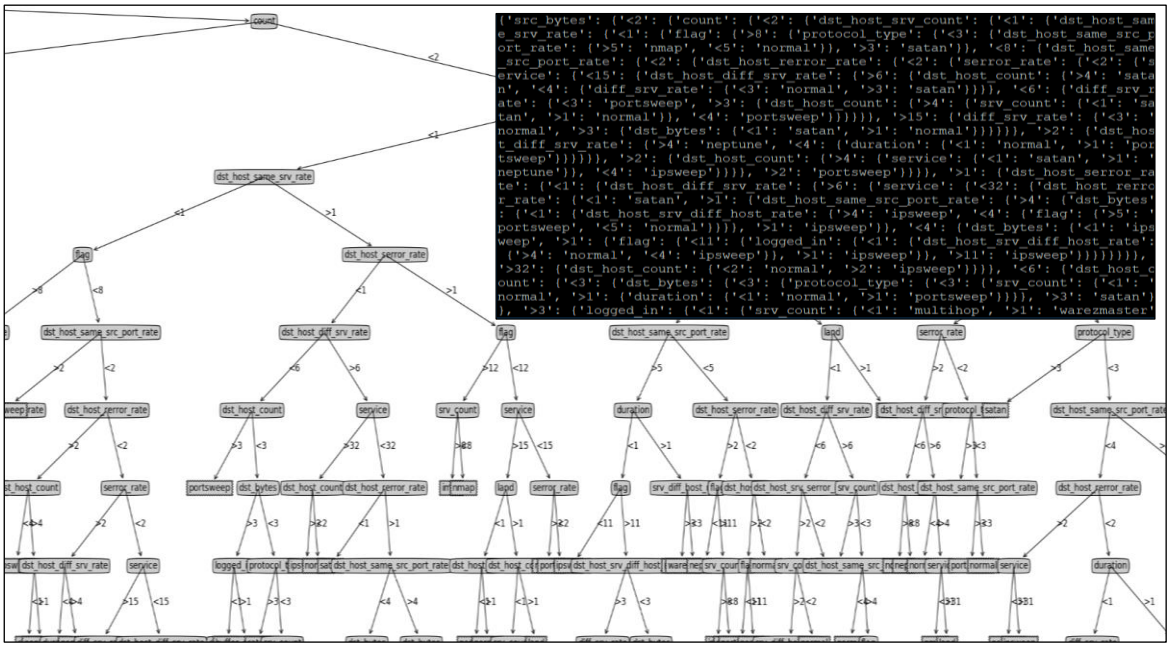
Tüm bunlara ek olarak, yapılan sınıflandırma işleminin neticesinde bilinmeyen saldırı olarak etiketlenen trafik kayıtları uzman görüşü için veri tabanında ilgili tabloya kopyalanır. Bu yöntem ile eğitim kümesinin yeni saldırı tipleri ile güncellenmesi sağlanır. Resim 4.32 üzerinde mevcut trafik kayıtların sınıflandırılması aşaması gösterilmektedir.



Resim 4.32. Gerçek trafik kayıtlarını sınıflandırma adımı

4.2.5. Bilginin sunumu adımı

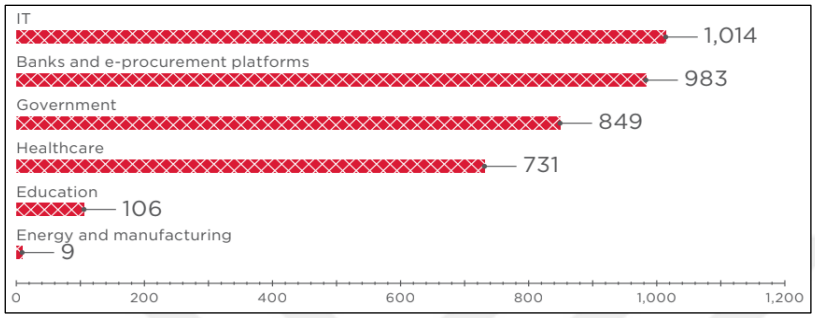
Veri madenciliği algoritmaları kullanılarak elde edilen örüntüler, işletim sistemi üzerinde bulunan dosyalara sözlük veri tipinde kaydedilir. Bu dosyalardan ilgili örüntüler okunarak sınıflandırma işlemleri gerçekleştirilir. İnsan tarafından, elde edilmiş örüntülerin kolaylıkla okunabilmesi için dosyalarda muhafaza edilen bilgiler karar ağacı yapısında resim olarak kullanıcıya sunulur. Bu kapsamda, CART algoritması kullanılarak dosyaya kaydedilmiş sözlük tipindeki örüntü ve örüntü üzerinden görselleştirilmiş karar ağacının küçük bir kesiti Resim 4.33 üzerinde görülebilir.



Resim 4.33. CART algoritması – ilgili karar ağacının örüntüsü ve kesiti

4.3. Önerilen Sistem İle Siber Tehdit İstihbaratı Üretme

Web sunucular ve web tabanlı uygulamalar oldukça popüler saldırı hedefleridir [210, 211]. Kamu, banka, sağlık, eğitim, üretim ve bilgi teknolojileri gibi sektörlerde kullanılan web uygulamaları oldukça yüksek sayıda saldırıya uğramaktadırlar [212]. Resim 4.34 üzerinde ilgili sektörlerde karşılaşılan günlük ortalama saldırı sayıları görülmektedir.



Resim 4.34. Sektörlere göre günlük ortalama web saldırıları [212]

Tüm bunlara ek olarak, literatür taramasında web uygulamalarının saldırganların doğrudan hedefi olmasa dahi ilgilerini çektiği görülmüştür [212]. Elde edilen bu bilgiler ışığında, internete açık bir sunucu çalıştırılmış ve web sunucusu olarak yapılandırılmıştır. Web sunucularında; FTP servisi sunucuya dosya yüklemek, SSH servisi sunucu üzerinde yönetimsel işlemleri gerçekleştirmek, veri tabanı servisi ise kayıt tutmak amacı ile kullanılabilir. TELNET servisi üzerinden ise birçok yönetimsel işlev gerçekleştirilebilir. Ayrıca HTTP, FTP, TELNET ve SSH gibi uygulama seviyesi protokoller sıklıkla tercih edilmekte ve kullanıcı trafiğinin büyük bir çoğunluğunu temsil etmektedir [213]. Bu sebeple, ilgili sunucunun gerçeği olabildiğince yansıtabilmesi ve saldırı yüzeyini mümkün olduğunca genişletebilmesi amacı ile FTP, TELNET ve SSH servisleri de sunucu üzerinde yapılandırılmıştır. Ayrıca PostgreSQL veri tabanı yönetim sistemi, web teknolojileri ile sıklıkla kullanıldığından dolayı tercih edilerek sunucuya entegre edilmiştir [214].

4.3.1. Veri toplama

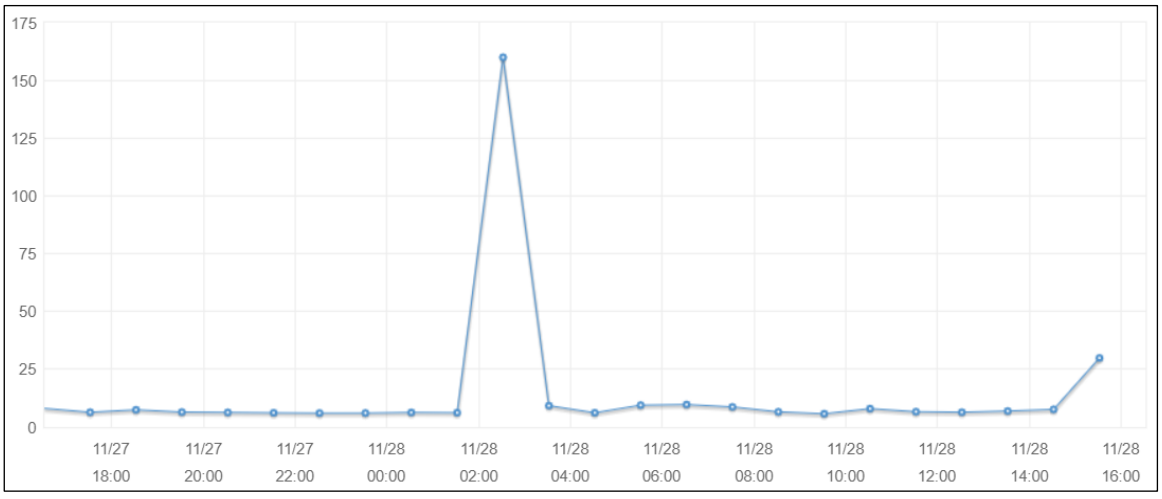
İlgili tüm servislerin zafiyetli gibi görülmesi ve kendisine yönlendirilen zararlı trafiği simüle edebilmesi için bal küpü sistemlerinden faydalanılmıştır. Birden fazla bal küpü sisteminin aynı anda çalışması amaçlandığından düşük etkileşimli bal küpü sistemleri tercih edilmiştir. Dionaea düşük etkileşimli bal küpü sistemi kullanılarak FTP, HTTP ve HTTPS servisleri gerçeklenmiştir. TELNET ve SSH için Cowrie, POSTGRESQL servisi için ise Sticky Elephant bal küpü yazılımları kullanılmıştır. Bal küpü sistemleri tarafından

üretilen günlük dosyaları ile disk alanının dolmaması için ilgili yazılımlarının günlük tutma işlevleri devre dışı bırakılmıştır. Ayrıca, internete açık sunucu için belirli bir IP ataması yapılmamıştır. Dolayısı ile belirli bir zaman periyodunda IP adresi değişmektedir. Böylece, saldırganların farklı bir sunucu ile karşılaştığı izlenimi verilmektedir. İlgili sunucuya gelen ve giden tüm trafik için izin verilmektedir. Sunucu, Amazon web servisleri ile bulut ortamında çalıştırılmıştır. İlgili sunucunun konumu Amerika Birleşik Devletleri Ohio eyaletinde bulunmaktadır. Sunucuya ait sistem özellikleri Çizelge 4.11'de gösterilmiştir.

Çizelge 4.11. İnternete açık sunucu sistem kaynakları

Kaynak	**Değer**
İşletim Sistemi	Ubuntu 16.04 Server
İşlemci Adedi	1 vCPU
Bellek Boyutu	1024 GB
Disk Alanı	8 GB

İnternet ortamında çalıştırılan ve bal küpü sistemleri barındıran ilgili sunucu vasıtası ile internet üzerinden gelen ve giden trafik tcpdump programı vasıtası ile kaydedilmiştir. Kaydedilen trafik Türkiye saati ile 27/11/2018 19:10 – 28/11/2018 19:10 tarih aralığına aittir. Resim 4.35 üzerinde sunucunun internete açılmasından itibaren ilk 24 saatlik süre için maruz kaldığı trafik gösterilmektedir. Her iki saatlik süreçler için ortalama paket bilgisi ilgili grafik üzerinde sunulmaktadır. Türkiye saati ile 19:10'da internete açılmış sunucu bilgileri saat farkından dolayı grafik üzerinde 16:10'da açılmış olarak görülmektedir.



Resim 4.35. İlk 24 saatlik periyot için gelen trafik paketleri ortalama sayısı

4.3.2. Siber tehdit istihbaratının üretilmesi

Kaydedilmiş trafik ağ trafiği dinleme modülü üzerine yerleştirilmiştir. Önerilen sistem ile ilgili trafik kayıtları sınıflandırılmıştır. Sınıflandırma işlemi için NSL-KDD Eğitim veri kümesi saldırı tipleri kullanılmış ve ilgili veri kümesi üzerinde yapılan karşılaştırmalar neticesinde daha iyi başarım elde etmiş olan ID3 algoritması tercih edilmiştir. Sınıflandırma işlemi neticesinde 740 adet farklı IP adresi üzerinden zararlı aktivitede bulunulduğu tespit edilmiştir. DOS saldırılarından neptune ve teardrop, Probing saldırılarından ise portsweep, nmap ve satan yakalanan saldırı tipleri arasındadır. Çizelge 4.12 üzerinde tespit edilmiş saldırılar hakkında detaylı bilgi verilmektedir. İlgili tabloda görüldüğü üzere, kaydedilmiş trafik üzerinden elde edilen toplam 5547 kayıt üzerinde yapılan sınıflandırma sonucunda nmap saldırısı 4 defa 2 farklı kaynaktan, portsweep saldırısı 2852 defa 715 farklı kaynaktan, satan saldırısı 146 defa 23 farklı kaynaktan, neptune saldırısı 20 defa 11 farklı kaynaktan ve teardrop saldırı ise 87 defa 24 farklı kaynaktan yapılmıştır. Tüm bunlara ek olarak, 35 benzersiz ip adresinin birden fazla saldırı gerçekleştirildiği tespit edilmiştir.

Çizelge 4.12 Yakalanmış trafik kayıtlarının sınıflandırma sonuçları

Grup	Tür	Toplam Sayı	Benzersiz IP adresi sayısı
Probing	nmap	4	2
Probing	portsweep	2852	715
Probing	satan	146	23
DOS	neptune	20	11
DOS	teardrop	87	24

Yapılan sınıflandırma işleminin ardından STİ üretilmiştir. İlgili kaydedilmiş trafik kullanılarak oluşturulmuş STİ, 3019 adet benzersiz kayıt barındırmaktadır. Böylece internet ortamına açılacak olan web sunucular için kullanılabilecek STİ elde edilmiştir. Ancak sunuculara yapılan saldırıların birçoğu benzer olduğundan ilgili STİ, internete açılacak her türde sunucu için kullanılabilirdir.

Tüm bunlara ek olarak, bazı tarama araçları yaptıkları analizler sonucunda ilgili servisin bir bal küpü sistemine ait olduğunu tespit edebilmektedir. Bu durumdan kaynaklı olarak saldırganların sunucu üzerinde bal küpü sistemi bulunduğunu fark etmesi ile saldırılarını ilerletmemeleri ve bu yüzden trafik kayıtlarında çoğunlukla tarama saldırısı kayıtları

bulunduğu düşünülmektedir. Ayrıca sunucunun internete açılmasından itibaren trafik alması, internetin sürekli olarak tarandığına işaret edebilir.

Geliştirilmiş modüllerin hazır sanal imajları, kaynak kodları ve önerilen sistem ile elde edilmiş STİ "https://github.com/arikansm/data-mining-based-cyber-threat-intelligence" adresinde paylaşılmıştır. Bu kapsamda, araştırmacılar tarafından bahsi geçen kaynaklara ilgili adres üzerinden ulaşılabilir.

5. SONUÇ VE ÖNERİLER

Bu çalışmada STİ üretmede yaşanan zorluklar analiz edilmiş, klasik STS yaklaşımları incelenmiş, veri madenciliği temelli STS'ler gözden geçirilmiş, STS tasarımında sıklıkla kullanılan veri kümeleri değerlendirilmiş, STİ standartları araştırılmış ve bal küpü teknolojileri incelenmiştir. Elde edilen bilgiler ışığında STİ üretmede yaşanan zorlukları en aza indirmek için 6 modülden oluşan sistemin tasarımı yapılmış ve uygulaması geliştirilmiştir. Ayrıca, önerilen sistemin uygulaması üzerinde örnek bir çalışma gerçekleştirilerek STIX formatında STİ üretilmiştir.

Çalışma kapsamında STİ üretmede yaşanan zorluklar;

- STİ üretmek için ilgili alan hakkında uzman bilgisine ihtiyaç duyulması, yeterli uzman bilgisine ulaşılamadığı durumlarda STİ elde edilememesi veya yanlış değerlendirilmiş bilgiler içerebilmesi,
- Araştırmacıların uzman bilgisine sahip olduğu durumlarda STİ üretme aşamalarının hemen hepsinin manuel işlemler içermesi ve bu sebepten dolayı oldukça uzun ve zahmetli bir çalışma sürecine ihtiyaç duyulması,
- STS'ler tarafından üretilen raporlar arasında genel bir benzerlik olmayışı, her STS'nin çıktısının farklı bir veri modeli barındırması, birden çok STS ile çalışılan durumlarda ise her bir STS raporunun veri modelinin iyi bilinmesine ihtiyaç duyulması ve bu sebeplerden dolayı farklı veri modellerinden kazanılan bilgiler ile STİ oluşturmanın zorlaşması
- STS'ler tarafından otomatik olarak üretilmediğinden dolayı kullanıcılar tarafından oluşturulmak zorunda olan STİ'nin, oldukça fazla iş yükü gerektirmesi ve bu sebepten dolayı birçok şirket ve/veya kurum tarafından bu iş için ayrılan zamanın gereksiz görülmesi veya zaman ayrılamaması olarak tanımlanmıştır.

Çalışmada, STİ oluşturmada kullanılan manuel adımların otomatize edilmesi ve uzman bilgisine ihtiyaç duyulmaksızın kolaylıkla STİ üretebilme yeteneğinin kullanıcılara kazandırılması amaçlanmıştır. Bu sebeple, çözüm yöntemi olarak veri madenciliği ile STİ üretilmesini kolaylaştıracak bir sistem önerilmiştir.

Önerilen sistem; kayıt modülü, raporlama modülü, saldırı tespit sistemi modülü, ağ trafiği dinleme modülü, zafiyet simülasyon modülü ve tehdit simülasyon modülü olmak üzere 6 adet modül ile geliştirilmiştir.

Modüller;

- STS için saldırı veri kümeleri, veri madenciliğinde kullanılan parametreler, STİ üretmek için kullanılan gerçek trafik kayıtları gibi bilgileri tutmak amacı ile kayıt modülü,
- STS'ye erişmek, bilinmeyen saldırı kayıtları için uzman görüşü almak, elde edilmiş gerçek trafik kayıtlarını görüntülemek, STİ üretmek ve oluşturulmuş STİ'lere erişmek için raporlama modülü,
- Veri madenciliği adımlarını gerçekleştirmek ve STS işlevleri yerine getirmek için saldırı tespit sistemi modülü,
- Canlı trafik bilgilerini yakalamak, yakalanmış canlı veya kaydedilmiş trafik bilgileri üzerinde gerçekleştirilen hesaplamalar ile sisteme uygun formatta trafik kayıtları elde etme amacıyla ağ trafiği dinleme modülü,
- Ham veri üretmek için saldırı yüzeyi oluşturma amacıyla zafiyet simülasyon modülü ve saldırı kayıtları üretme amacıyla tehdit simülasyon modülü kullanılmıştır.

Geliştirilen önerilen sistemin uygulaması üzerinde, KDD CUP 99 %10 veri kümesi ve NSL-KDD Eğitim veri kümesi kullanılarak tabakalı 10 katlamalı çapraz doğrulama gerçekleştirilmiştir. İlgili sonuçlar literatürdeki benzer çalışmalar ve veri madenciliğinde güvenilirliğini kanıtlamış Weka aracı sonuçları ile kıyaslanmıştır. Kıyaslama neticeleri ile önerilen sistemin veri madenciliği için başarılı bir şekilde kullanılabileceği görülmüştür.

Önerilen sistemin uygulması için örnek bir çalışma olarak, 24 saat boyunca internete açık çalıştırılan bir sunucu ile bal küpü sistemleri kullanılarak saldırganların dikkati çekilmiş ve ilgili trafik kayıtları kaydedilmiştir. Kaydedilmiş trafik kayıtları, dinleme modülü üzerine yerleştirilmiş ve önerilen sistemin ilgili kayıtlar üzerinden STİ oluşturması sağlanmıştır. Örnek çalışma ile önerilen sistemin STİ üretmek için başarılı bir şekilde kolaylıkla kullanılabileceği gösterilmiş ve internet ortamına açılacak olan sunucular için kullanılabilecek STİ üretilmiştir.

Çalışmada gerçek trafik kayıtlarının sınıflandırılması için saldırı tiplerinin öğrenilmesi amacı ile kullanılan NSL-KDD veri kümesinin barındırdığı bazı saldırı tiplerinde yıllar içinde değişiklikler gözlenebilir. Bu sebeple, NSL-KDD veri kümesinde bulunan saldırıların güncelliğinin test edilmesine ihtiyaç duyulmuştur. Fakat, NSL-KDD Eğitim veri kümesinde bulunan tüm saldırı tiplerinin güncel olduğunun ispatlanması oldukça

zahmetli ve kapsamlı bir çalışma gerektirdiğinden bu çalışma kapsamında yalnızca portsweep saldırısı ele alınmıştır. İlgili saldırı için NSL-KDD Eğitim veri kümesinin güncel olduğu ispatlanmış ve bu sebeple çalışmada diğer saldırı tiplerinin de güncel olduğu varsayılmıştır. Ayrıca, ilgili saldırı kayıtları üzerinde gerçekleştirilen incelemeler neticesinde STS'nin başarımının, incelenen saldırı trafiğinin boyutu ve gerçekleşme hızı ile doğru orantılı olduğu sonucuna da ulaşılmıştır.

Çalışmanın sınırlılıkları;

- Önerilen sistem tarafından üretilen STİ'nin kapsamının yalnızca ağ trafiğinden oluşturulabilir teknik STİ ile sınırlı olması,
- Saldırganların dikkatini çekmek için kullanılan bal küpü sistemlerinin bazı tarama araçları vasıtası ile fark edilebilir olmasından dolayı ilgili sistemin bal küpü olduğunu anlayan kötü niyetli kullanıcının saldırıyı ilerletmeyebileceği,
- İnternete açık olarak çalıştırılan sunucu için alınan bulut hizmetinin bazı saldırı trafiğini ve/veya saldırıları engellemiş olabileceği,
- Kullanılan sanallaştırma teknolojisinden kaynaklı performans kayıplarından dolayı geliştirilmiş modüllerde bazı işlemlerin daha yavaş gerçekleşmiş olabileceği,
- Geliştirilmiş STS'nin şifrelenmiş paketlerin içeriğini çözemeyeceğinden dolayı ilgili paketlerin taşıdığı verileri analiz edememesi,
- Ağ paketlerinin değiştirilebilir ve manipüle edilebilir olmasından dolayı trafik paketleri üzerinde gerçekleştirilen hesaplamaların yanıltılabilir olması ve bu sebeple yanlış alarm oluşabilmesi veya hiç alarm oluşmaması gibi tanımlanabilir.

İlgili sınırlılıklara ek olarak, ele alınmış çalışma ile yalnızca IPV4 adresleri yakalabilmekte ve ilgili versiyona ait gerçek trafik kayıtları incelenebilmektedir. İlerleyen çalışmalarda ilgili sistemin IPV6 ile çalışması sağlanarak bu versiyona ait gerçek trafik kayıtlarının da incelenmesi mümkün kılınabilir. Ayrıca, literatürde sıklıkla tercih edilen KDD CUP 99 ve NSL-KDD veri kümeleri üzerinden öğrenilen saldırı türleri ile STİ üretilebileceği gösterilmiştir. Ancak çalışmada, önerilen sistemin güncel ve günümüz saldırıları öğrenebilmesi için kullanılabilecek yeni bir veri kümesine ihtiyaç duyulmuştur. İlerleyen çalışmalarda, ihtiyaç duyulan veri kümesinin sağlanması ile hem önerilen sistem daha etkili bir şekilde kullanılabilir hem de literatüre yeni ve güncel bir saldırı veri kümesi ile katkı sağlanabilir. Tüm bunlara ek olarak, araştırmacılar sonraki çalışmalarda farklı veya

yeni veri kümelerini, veri madenciliği algoritmalarını, STİ formatlarını kullanarak önerilen sistemin performansını geliştirebilir.

STİ üretmek amacıyla gerçekleştirilmiş Eric Nunes ve arkadaşları tarafından ulaşılabilir olmayan gizli internet kaynaklarından elde ettikleri bilgiler üzerine veri madenciliği tekniklerinin uygulanması gibi çalışmalar incelendiğinde elde edilmiş istihbarat bilgilerinin herhangi bir STİ standardına uygun olarak üretilmediği ve STİ üretmek için gerçek trafik kayıtlarından faydalanılmadığı görülmüştür. STS geliştirmek için gerçekleştirilmiş Dikshant Gupta ve arkadaşlarının veri madenciliği algoritmalarını kullanarak ağ paketlerini sınıflandırmak için kural üreten çalışması ile Shankar M. Banik ve arkadaşlarının ağ trafiğini izlemek yerine işletim sistemi günlüklerini analiz ederek saldırı tiplerini yakalayan çalışması gibi çalışmalar analiz edildiğinde STS ile STİ üretilmesi için yapılmış bir çalışmaya rastlanmamıştır. Dolayısıyla bu çalışmada üretilen STİ için kaynak olarak gerçek trafik kayıtları kullanıldığından ve bu kapsamda saldırı tiplerinin öğrenilmesi ve trafiğin sınıflandırılması için STS'den faydalanıldığından, çalışma ile ilgili yakın veya benzer araştırma bulunamamıştır. Tüm bu bilgilere ek olarak, gerçekleştirilen bu araştırmanın bildiğimiz kadarı ile literatürdeki benzer çalışmalardan farkı;

- STIX formatında STİ oluşturmak için gerçek trafik kayıtları üzerinde incelemeler yapan veri madenciliği temelli bir STS'den faydalanılmış olması,
- KDD CUP 99 ve NSL-KDD veri kümesi kullanılarak canlı ve/veya kaydedilmiş gerçek trafik kayıtları üzerinde karar ağaçları algoritmalarından faydalanarak analizler gerçekleştirebilecek yetkinlikte bir STS geliştirilmesi,
- NSL-KDD veri kümesinde bulunan saldırı türlerinden araştırmacının bilgisi doğrultusunda seçilen portsweep saldırısı için güncelliğin test edilmiş olması,
- STIX formatında STİ üretme adımlarını kolaylaştırmak için gerçek trafik kayıtlarını kullanabilen bir sistemin önerilmiş, uygulamasının geliştirilmiş, geliştirilen uygulama üzerinde ise örnek çalışmasının yapılmış olmasıdır.

Çalışma kapsamında önerilen sistem şirketler, kamu kurumları ve bireyler tarafından kullanılması ile STİ'nin kolaylıkla üretimi sağlanabilir. Böylece siber güvenliğin sağlanması için büyük öneme sahip olan STİ, birçok paydaş arasında dağıtılarak saldırılardan korunmak için bilgi paylaşımı sağlanabilir. Büyük ölçekli şirketler veya kamu kurumları tek bir noktadan STİ üreterek ilgili alt gruplarını saldırılar hakkında bilgilendirebilir ve korunması için gerekli bilgileri sunabilir. Çalışma kapsamında üretilmiş

STİ üzerinden internet ortamında çalıştırılacak olan sunucuların karşılaşabilecekleri saldırı tipleri ve saldırganlar hakkında bilgi elde edilebilir. Bu bilgiler ışığında sistem yöneticisi tarafından gerekli önlemlerin alınmasıyla ilgili sunucular için bahsi geçen saldırılar gerçekleşmeden engellenebilir.

KAYNAKLAR

1. İnternet: Planqué, D. Cyber Threat Intelligence From confusion to clarity; An investigation into Cyber Threat Intelligence. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fopenaccess.leidenuniv.nl%2Fbitstream%2Fhandle%2F1887%2F64551%2FPlanque_D_2017_CS.pdf%3Fsequence%3D1&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

2. Barnum, S. (2012). Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX). *MITRE Corporation*, 11, 1-22.

3. Dehghantanha, A., Conti, M. and Dargahi, T. (Eds.). (2018). Cyber threat intelligence. *Springer International Publishing.*

4. İnternet: The UK's National Cyber Security Centre. National Cyber Security Strategy 2016-2021. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fassets.publishing.service.gov.uk%2Fgovernment%2Fuploads%2Fsystem%2Fuploads%2Fattachment_data%2Ffile%2F567242%2Fnational_cyber_security_strategy_2016.pdf+&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

5. Chismon, D. and Ruks, M. (2015). Threat intelligence: Collecting, analysing, evaluating. *MWR InfoSecurity Ltd*.

6. Rattan, A., Kaur, N., Chamotra, S. and Bhushan, S. (2017). Attack Data Usability and Challenges in its Capturing and Sharing. *International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE).*

7. İnternet: Impe, K. V. Comparing Different Tools for Threat Sharing. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fsecurityintelligence.com%2Fcomparing-different-tools-for-threat-sharing%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

8. Nunes, E., Diab, A., Gunn, A., Marin, E., Mishra, V., Paliath, V., Robertson, J., Shakarian, J., Thart, A. and Shakarian, P. (2016). *Darknet and deepnet mining for proactive cybersecurity threat intelligence*. In 2016 IEEE Conference on Intelligence and Security Informatics (ISI), 7-12.

9. Rowland, C. H. (2002). *U.S. Patent No. 6,405,318*. Washington, DC: U.S. Patent and Trademark Office

10. Scheidell, M. (2009). *U.S. Patent No. 7,603,711*. Washington, DC: U.S. Patent and Trademark Office.

11. Hong, L. (2009). *Immune mechanism based intrusion detection systems*. IEEE In Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC'09. International Conference on, 568-571.

12. Depren, O., Topallar, M., Anarim, E. and Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications*, 29(4), 713-722.

13. Portnoy, L. (2000). *Intrusion detection with unlabeled data using clustering* (Doctoral dissertation, Columbia University), USA.

14. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2), 18-28.

15. Om, H. and Kundu, A. (2012). *A hybrid system for reducing the false alarm rate of anomaly intrusion detection system*. IEEE In Recent Advances in Information Technology (RAIT), 2012 1st International Conference on, 131-136.

16. Landge, M. R. S. and Wadhe, M. A. P. (2013). Review of Various Intrusion Detection Techniques based on Data mining approach. *International Journal of Engineering Research and Applications (IJERA)*. 3(3), 430-435

17. Elekar, K. S. (2015). *Combination of data mining techniques for intrusion detection system.* IEEE In Computer, Communication and Control (IC4), 2015 International Conference on, 1-5.

18. Goeschel, K. (2016). *Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis*. IEEE In SoutheastCon *2016*, 1-6.

19. Chang, C. C. and Lin, C. J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), 27.

20. Ng, J., Joshi, D. and Banik, S. M. (2015). *Applying data mining techniques to intrusion detection*. IEEE In 2015 12th International Conference on Information Technology-New Generations (ITNG), 800-801.

21. Gupta, D., Singhal, S., Malik, S. and Singh, A. (2016). *Network intrusion detection system using various data mining techniques*. IEEE In Research Advances in Integrated Navigation Systems (RAINS), International Conference on, 1-6.

22. İnternet: From character to personality. NSL-KDD dataset. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.unb.ca%2Fcic%2Fdatasets%2Fnsl.html+&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

23. El Moussaid, N. and Toumanari, A. (2014). *Overview of intrusion detection using data-mining and the features selection*. IEEE In Multimedia Computing and Systems (ICMCS), 2014 International Conference on, 1269-1273.

24. Sağıroğlu, Ş., Yolaçan, E. N. ve Yavanoğlu, U. (2011). Zeki Saldırı Tespit Sistemi Tasarımı ve Gerçekleştirilmesi. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 26(2). 325-340

25. Subaira, A. S. and Anitha, P. (2014). *Efficient classification mechanism for network intrusion detection system based on data mining techniques: a survey*. IEEE In Intelligent Systems and Control (ISCO), 2014 IEEE 8th International Conference on, 274-280.

26. Arıkan, S. M. ve Benzer, R. Bir Güvenlik Trendi: Bal Küpü. *Acta INFOLOGICA*, 2(1), 1-11.

27. Singh, A. N. and Joshi, R. C. (2011). *A honeypot system for efficient capture and analysis of network attack traffic.* IEEE In Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on, 514-519.

28. Rødfoss, J. T. (2011). *Comparison of open source network intrusion detection systems*, Yüksek Lisans Tezi, University of Oslo, Oslo.

29. İnternet: Jbanderson, A. How to set up Dionaea Honeypot to evade Nmap detection. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.attacusatlas.com%2Fhow-to-set-up-dionaea-honeypot-to-evade-nmap-detection%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

30. Larose, D. T. and Larose, C. D. (2014). Discovering knowledge in data: an introduction to data mining. *John Wiley & Sons*

31. Han, J., Pei, J. and Kamber, M. (2011). Data mining: concepts and techniques. *Elsevier*.

32. Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3), 37.

33. Fayyad, U. and Stolorz, P. (1997). Data mining and KDD: Promise and challenges. *Future Generation Computer Systems*, 13(2-3), 99-115.

34. Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11), 27-34.

35. Fayyad, U. M., Piatetsky-Shapiro, G. and Smyth, P. (1996). *Knowledge Discovery and Data Mining: Towards a Unifying Framework.* KDD'96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 82-88.

36. Azevedo, A. I. R. L. and Santos, M. F. (2008). *KDD, SEMMA and CRISP-DM: a parallel overview*. IADIS European Conference Data Mining 2008, 182-185

37. Maimon, O., and Rokach, L. (2009). Introduction to knowledge discovery and data mining. *In Data Mining and Knowledge Discovery Handbook*, 1-15, Springer, Boston, MA.

38. Norton, M. J. (1999). Knowledge discovery in databases. *Library Trends*, 48(1), 9.

39. Özen, Ş. (2000). Türk yönetim/organizasyon yazininda yöntem sorunu: Kongre bildirileri üzerine bir inceleme. *Doğu Akdeniz Üniversitesi Turizm Arastirmalari Dergisi*, 1(1), 89-118.

40. Yaşın, C. (2003). Siyasal Araştırmalarda Örneklem Sorunu. *İletişim Dergisi*, 18, 147-172.

41. Özdemir, A. (2008). Yönetim Biliminde İleri Araştirma Yöntemleri ve Uygulamalar. *Beta Yayınları*.

42. Bernard, H. R. and Bernard, H. R. (2012). Social research methods: Qualitative and quantitative approaches. *Sage*.

43. Karamustafaoğlu, S., Çağlak, A. ve Meşeci, B. (2012). Alternatif Ölçme Değerlendirme Araçlarina İlişkin Sinif Öğretmenlerinin Öz Yeterlilikleri. *Amasya Üniversitesi Eğitim Fakültesi Dergisi*, 1(2), 167-179.

44. Kırtay, S. H., Ekmekçi, N., Halıcı, T., Ketenci, U., Aktas, M. S. ve Kalıpsız, O. (2015). *Pazar Sepeti Analizi için Orneklem Olusturulması ve Birliktelik Kurallarının Cıkartılması*, 9. Ulusal Yazılım Mühendisliği Sempozyumu, 172-183.

45. Wu, X., Zhu, X., Wu, G. Q. And Ding, W. (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 26(1), 97-107.

46. Baykal, A. (2006). Veri Madenciliği ve Uygulama Alanları. *DÜ Ziya Gökalp Eğitim Fakültesi Dergisi*, 7, 95-107.

47. Sever, H., ve Buket, O. (2003). Veri Tabanlarında Bilgi Keşfine Formel Bir Yaklaşım Kısım I: Eşleştirme Sorguları ve Algoritmalar. *Bilgi Dünyası*, 3(2), 173-204.

48. Savaş, S., Topaloğlu, N. ve Yılmaz, M. (2012). Veri madenciliği ve Türkiye'deki uygulama örnekleri, *İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi*, 21, 1-2.

49. Vahaplar, A. ve İnceoğlu, M. M. (2001). *Veri madenciliği ve elektronik ticaret*. Türkiye'de İnternet Konferansları, 1-3.

50. Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387.

51. Zaïane, O. R., Han, J., Li, Z. N. and Hou, J. (1998, November). Mining multimedia data. *In Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research*, 24, IBM Press.

52. Zhuang, Y. T., Yang, Y. and Wu, F. (2008). Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval. *IEEE Transactions on Multimedia*, 10(2), 221-229.

53. Bhatt, C. A. and Kankanhalli, M. S. (2011). Multimedia data mining: state of the art and challenges. *Multimedia Tools and Applications*, 51(1), 35-76.

54. Ozekes, S. (2003). Veri Madenciligi Modelleri ve Uygulama Alanlari. *Istanbul Ticaret University Journal*, 3, 65-82.

55. Kantardzic, M. (2011). Data mining: concepts, models, methods, and algorithms. *John Wiley & Sons.*

56. Nizam, H. ve Akın, S. S. (2014). *Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması. XIX.* Türkiye'de İnternet Konferansı.

57. İnternet: From character to personality. Data Mining Concepts Classification. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdocs.oracle.com%2Fcd%2FB28359_01%2Fdatamine.111%2Fb28129%2Fclassify.htm%23DMCON004&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

58. İnternet: From character to personality. Data Mining Concepts Regression. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fdocs.oracle.com%2Fcd%2FB28359_01%2Fdatamine.111%2Fb28129%2Fregress.htm%23DMCON005&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

59. İnternet: Chapple, M. Use Regression to Predict Where You Might Live. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.lifewire.com%2Fregression-1019655&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

60. Akyol, K., Şen, B. ve Çalık, E. (2012). *Biyokimya ve Hemogram Laboratuvar Test Sonuçlarının Lojistik Regresyon Yöntemiyle Analizi*. *A*kademik Bilişim'12 - XIV. Akademik Bilişim Konferansı Bildirileri, 311-321

61. Tolga, K. ve Hünkar, T. (2018). Medya İşletmelerinde Yeni Nesil Pazarlama: Freemium Spotify Üzerine Bir Araştırma. *The Turkish Online Journal of Design Art and Communication*, 8(4), 684-693.

62. Itskovich, L. and Kuznetsov, S. (2011). Machine learning methods in character recognition. *In International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, 322-329, Springer, Berlin, Heidelberg.

63. Gupta, R. and Aggarwal, G. (2016). *Human speech sentiments recognition: A data mining approach for categorization of speech*. IEEE In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on, 3987-3991.

64. Jadon, E. and Sharma, R. (2017). Data Mining: Document Classification using Naive Bayes Classifier. *International Journal of Computer Applications*, 167(6), 13-16.

65. Akpınar, H. (2000). Veri tabanlarında bilgi keşfi ve veri madenciliği. *İÜ İşletme Fakültesi Dergisi*, 29(1), 1-22.

66. Qin, B., Xia, Y. and Li, F. (2009). *DTU: a decision tree for uncertain data*. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, 4-15

67. Sun, W., Chen, J. and Li, J. (2007). Decision tree and PCA-based fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, 21(3), 1300-1317.

68. Hssina, B., Merbouha, A., Ezzikouri, H. and Erritali, M. (2014). A comparative study of decision tree ID3 and C4. 5. *International Journal of Advanced Computer Science and Applications*, 4(2).

69. Venables, W. N. and Ripley, B. D. (2002). *Tree-based methods*. In Modern Applied Statistics with S, 251-269.

70. Peng, W., Chen, J. and Zhou, H. (2009). An Implementation Of ID3-Decision Tree Learning Algorithm. *School of Computer Science & Engineering, University of New South Wales, Sydney, Australia*.

71. Slocum, M. (2012). Decision making using id3 algorithm. *Insight: River Academic J*, 8(2).

72. Bahety, A. (2014). Extension and evaluation of id3–decision tree algorithm. *Entropy (S)*, 2(1).

73. Bhardwaj, R. and Vatta, S. (2013). Implementation of ID3 algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6).

74. Shao, X., Zhang, G., Li, P. and Chen, Y. (2001). Application of ID3 algorithm in knowledge acquisition for tolerance design. *Journal of materials processing technology*, 117(1-2), 66-74.

75. Soni, S. (2010). Implementation of multivariate data set by CART algorithm. *Journal of Information Technology and Knowledge Management*, 2(2), 455-459.

76. Sathyadevi, G. (2011). *Application of CART algorithm in hepatitis disease diagnosis*. IEEE In Recent Trends in Information Technology (ICRTIT), 2011 International Conference on, 1283-1287.

77. Chrysos, G., Dagritzikos, P., Papaefstathiou, I. and Dollas, A. (2013). HC-CART: A parallel system implementation of data mining classification and regression tree (CART) algorithm on a multi-FPGA system. *ACM Transactions on Architecture and Code Optimization (TACO)*, 9(4), 47.

78. Santhanam, T. and Sundaram, S. (2010). Application of CART algorithm in blood donors classification. *Journal of computer Science*, 6(5), 548.

79. Mola, F. and Siciliano, R. (1997). A fast splitting procedure for classification trees. *Statistics and Computing*, 7(3), 209-216.

80. Rutkowski, L., Jaworski, M., Pietruczuk, L. and Duda, P. (2014). The CART decision tree for mining data streams. *Information Sciences*, 266, 1-15.

81. Li, M. (2017). *Application of CART decision tree combined with PCA algorithm in intrusion detection*. IEEE In 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 38-41.

82. Rahman, C. M., Farid, D. M., Harbi, N., Bahri, E. and Rahman, M. Z. (2010). Attacks classification in adaptive intrusion detection using decision tree. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, 4(3), 368-372

83. İnternet: Hong, K. Machine Learning with scikit-learn. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.bogotobogo.com%2Fpython%2Fscikit-learn%2Fscikt_machine_learning_Decision_Tree_Learning_Informatioin_Gain_IG_Impurity_Entropy_Gini_Classification_Error.php&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

84. İnternet: Grimaldi, E. Decision Tree: An algorithm that works like the human brain. URL: http://www.webcitation.org/query?url=https%3A%2F%2Ftowardsdatascience.com%2Fdecision-tree-an-algorithm-that-works-like-the-human-brain-8bc0652f1fc6&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

85. Rumelhart, D. E., Widrow, B. and Lehr, M. A. (1994). The basic ideas in neural networks. *Communications of the ACM*, 37(3), 87-93.

86. Tokar, A. S. and Johnson, P. A. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3), 232-239.

87. Patra, J. C., Pal, R. N., Chatterji, B. N. and Panda, G. (1999). Identification of nonlinear dynamic systems using functional link artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part b (cybernetics)*, 29(2), 254-262.

88. Bhama, S. and Singh, H. (1993). Single layer neural networks for linear system identification using gradient descent technique. *IEEE Transactions on Neural Networks*, 4(5), 884-888.

89. Specht, D. F. (1990). Probabilistic neural networks. *Neural networks*, 3(1), 109-118.

90. Hartman, E. J., Keeler, J. D. and Kowalski, J. M. (1990). Layered neural networks with Gaussian hidden units as universal approximations. *Neural Computation*, 2(2), 210-215.

91. Maren, A. J. (1991, February). *A logical topology of neural networks*. Proceedings of the Second Workshop in Neural Networks, 1-27

92. İnternet: Uslu, M. Yapay Sinir Ağları (YSA) Nedir ?. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fkod5.org%2Fyapay-sinir-aglari-ysa-nedir%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

93. Heaton, J. (2010). Programming neural networks with Encog 2 in Java. *Heaton Research, Inc, USA*.

94. Garcia, V., Debreuve, E. and Barlaud, M. (2008). *Fast k nearest neighbor search using GPU*. In 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 1-6.

95. Wang, J., Neskovic, P. and Cooper, L. N. (2007). Improving nearest neighbor rule with a simple adaptive distance measure. *Pattern Recognition Letters*, 28(2), 207-213.

96. Baldini, G., Steri, G., Giuliani, R. and Gentile, C. (2017). *Imaging time series for internet of things radio frequency fingerprinting*. IEEE In 2017 International Carnahan Conference on Security Technology (ICCST), 1-6.

97. Ahmad, A. and Dey, L. (2007). A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters*, 28(1), 110-118.

98. Taşcı, E. ve Onan, A. (2016). K-en yakın komşu algoritması parametrelerinin sınıflandırma performansı üzerine etkisinin incelenmesi. *Akademik Bilişim*.

99. Duda, R. O., Hart, P. E. and Stork, D. G. (2012). Pattern classification. *John Wiley & Sons*.

100. Matsumoto, M. and Hori, J. (2014). Classification of silent speech using support vector machine and relevance vector machine. *Applied Soft Computing*, 20, 95-102.

101. Kuzey, C. ve Öner, E. (2012). *Veri madenciliğinde destek vektör makinaları ve karar ağaçları yöntemlerini kullanarak bilgi çalışanlarının kurum performansı üzerine etkisinin ölçülmesi ve bir uygulama*, Doktora Tezi, İstanbul Üniversitesi, İstanbul

102. Chen, S., Härdle, W. K. and Moro, R. (2006). Estimation of default probabilities with support vector machines. *Discussion Paper 77, SFB 649 Humboldt University, Berlin*.

103. Meyer, D. and Wien, F. T. (2001). Support vector machines. *R News*, 1(3), 23-26.

104. Friedman, J., Hastie, T. and Tibshirani, R. (2001). The elements of statistical learning. *Springer Series in Statistics*.

105. Kavzoğlu, T. ve Çölkesen, İ. (2010). Destek vektör makineleri ile uydu görüntülerinin sınıflandırılmasında kernel fonksiyonlarının etkilerinin incelenmesi. *Harita Dergisi*, 144(7), 73-82.

106. Leung, K. M. (2007). Naive bayesian classifier. *Polytechnic University Department of Computer Science/Finance and Risk Engineering*.

107. Rish, I. (2001). *An empirical study of the naive Bayes classifier*. In IJCAI 2001 workshop on empirical methods in artificial intelligence,41-46.

108. Shah, M. and Nair, S. (2015). A survey of data mining clustering algorithms. *International Journal of Computer Applications*, 128(1), 1-5.

109. Huang, A. (2008). Similarity measures for text document clustering. In Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), 49-56.

110. Berkhin, P. (2006). *A survey of clustering data mining techniques*. In Grouping Multidimensional Data, 25-71.

111. Jianliang, M., Haikun, S. and Ling, B. (2009). *The application on intrusion detection based on k-means cluster algorithm*. In Information Technology and Applications, 2009. IFITA'09. International Forum on, 150-152.

112. Verma, M., Srivastava, M., Chack, N., Diswar, A. K. and Gupta, N. (2012). A comparative study of various clustering algorithms in data mining. *International Journal of Engineering Research and Applications (IJERA)*, 2(3), 1379-1384.

113. Yeşilbudak, M., Kahraman, H. T. ve Karacan, H. (2011). Veri madenciliğinde nesne yönelimli birleştirici hiyerarşik kümeleme modeli. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 26(1), 27-39.

114. Öztürk, M. B. Ve Can, B. (2016). *Clustering word roots syntactically*. IEEE In Signal Processing and Communication Application Conference (SIU), 1461-1464.

115. Eyüpoğlu, C., Aydın, M. A., Sertbaş, A., Zaim, A. H. ve Onur, Ö. (2017). Büyük Veride Kişi Mahremiyetinin Korunması. *Bilişim Teknolojileri Dergisi*, 10(2), 177-184.

116. İnternet: Akın, Ç. E. Yığınsal (Agglomerative) Kümeleme – #4. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fcagriemreakin.com%2Fveri-bilimi%2Fagglomerative-clustering-4.html&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

117. Davidson, I. and Ravi, S. S. (2005). *Agglomerative hierarchical clustering with constraints: Theoretical and empirical results*. In European Conference on Principles of Data Mining and Knowledge Discovery, 59-70.

118. Sharma, A., López, Y. and Tsunoda, T. (2017). Divisive hierarchical maximum likelihood clustering. *BMC Bioinformatics*, 18(16), 546.

119. Patnaik, A. K., Bhuyan, P. K. and Rao, K. K. (2016). Divisive Analysis (DIANA) of hierarchical clustering and GPS data for level of service criteria of urban streets. *Alexandria Engineering Journal*, 55(1), 407-418.

120. Kriegel, H. P., Kröger, P., Sander, J. and Zimek, A. (2011). Density- based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3), 231-240.

121. Ankerst, M., Breunig, M. M., Kriegel, H. P. and Sander, J. (1999). *OPTICS: ordering points to identify the clustering structure*. In ACM Sigmod record, 49-60.

122. Ma, S., Wang, T., Tang, S., Yang, D., and Gao, J. (2003, August). A new fast clustering algorithm based on reference and density. *In International Conference on Web-Age Information Management*, 214-225, Springer, Berlin, Heidelberg.

123. Kailing, K., Kriegel, H. P., and Kröger, P. (2004). *Density-connected subspace clustering for high-dimensional data*. In Proceedings of the 2004 SIAM International Conference on Data Mining, 246-256.

124. Thakare, M. A. T. and Chaudhari, M. S. (2014). A Survey on Framework for Improved Web Data Clustering Using Language Processing Technique. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(12), 3941-3944.

125. Mann, A. K. and Kaur, N. (2013). Grid density based clustering algorithm. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(6).

126. Hua, Z., Zhenxing, W., Liancheng, Z. and Qian, W. (2010). *Clustering algorithm based on characteristics of density distribution*. IEEE In Advanced Computer Control (ICACC), 2010 2nd International Conference on, 431-435.

127. Özçınar, H. (2006). *KPSS sonuçlarının veri madenciliği yöntemleriyle tahmin edilmesi,* Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli.

128. Tate, A. and Bewoor, L. (2017). *Survey on Frequent Pattern Mining Algorithm for Kernel Trace.* IEEE In Advance Computing Conference (IACC), 2017 IEEE 7th International, 793-798.

129. Güngör, E., Yalçın, N. and Yurtay, N. (2013). *Apriori algoritması ile teknik seçmeli ders seçim analizi*. UZEM 2013 Ulusal Uzaktan Eğitim ve Teknolojileri Sempozyumu, 01-03.

130. Hipp, J., Güntzer, U. and Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1), 58-64.

131. Leung, C. K. S. (2011). Mining uncertain data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4), 316-329.

132. Baker, R. S. J. D. (2010). Data mining for education. *International Encyclopedia of Education*, 7(3), 112-118.

133. İnternet: From character to personality. Data Mining Applications & Trends. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.tutorialspoint.com%2Fdata_mining%2Fdm_applications_trends.htm&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

134. İnternet: From character to personality. Walmart. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fcorporate.walmart.com%2Four-story&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

135. Lichtenstein, N. (Ed.). (2006). Wal-Mart: The face of twenty-first-century capitalism. *The New Press*.

136. İnternet: Ian Davey and Technolegis. Consumers, Big Data, and Online Tracking in the Retail Industry: A CASE STUDY OF WALMART. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fcenterformediajustice.org%2Fresources%2Fconsumers-big-data-and-online-tracking-in-the-retail-industry-a-case-study-of-walmart%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

137. İnternet: From character to personality. Overwatch, a Team-Based Shooter. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fplayoverwatch.com%2Fen-us%2Fgame%2Foverview%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

138. Braun, P., Cuzzocrea, A., Keding, T. D., Leung, C. K., Padzor, A. G. and Sayson, D. (2017). Game data mining: clustering and visualization of online game data in cyber-physical worlds. *Procedia Computer Science*, 112, 2259-2268.

139. Superby, J. F., Vandamme, J. P. and Meskens, N. (2006). *Determination of factors influencing the achievement of the first-year university students using data mining methods*. In Workshop on Educational Data Mining, 234.

140. Li, S. and Luo, Y. (2010). *High performance flow feature extraction with multi-core processors.* IEEE In Networking, Architecture and Storage (NAS), 2010 IEEE Fifth International Conference on, 193-201.

141. İnternet: Pratt, M. K. What is an intrusion detection system (IDS)? A valued function with significant challenges. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.csoonline.com%2Farticle%2F3255632%2Fwhat-is-an-intrusion-detection-system-how-an-ids-spots-threats.html&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

142. Yeo, L. H., Che, X. and Lakkaraju, S. (2017). Modern Intrusion Detection Systems. *arXiv preprint arXiv:1708.07174*.

143. Sarmah, A. (2001). Intrusion detection systems: Definition, need and challenges. *White Paper from SANS Institute*.

144. Warzyński, A. and Kołaczek, G. (2018). *Intrusion detection systems vulnerability on adversarial examples*. IEEE In 2018 Innovations in Intelligent Systems and Applications (INISTA), 1-4.

145. Debar, H., Dacier, M. and Wespi, A. (2000). *A revised taxonomy for intrusion-detection systems*. In Annales des télécommunications, 361-378.

146. Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A. And Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications*, 36(1), 42-57.

147. Jaswal, K., Kumar, P. and Rawat, S. (2015). *Design and development of a prototype application for intrusion detection using data mining*. IEEE In Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), 2015 4th International Conference on, 1-6.

148. Liao, H. J., Lin, C. H. R., Lin, Y. C. and Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36(1), 16-24.

149. İnternet: Redemacher, L. The Disadvantages of Intrusion Detection Systems. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.techwalla.com%2Farticles%2Fthe-disadvantages-of-intrusion-detection-systems&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

150. İnternet: From character to personality. The Pros & Cons of Intrusion Detection Systems. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fblog.rapid7.com%2F2017%2F01%2F11%2Fthe-pros-cons-of-intrusion-detection-systems%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

151. Vijay, P. (2017). Performance Evaluation of Classification Techniques for Intrusion Detection in Noisy Datasets. *International Journal on Recent and Innovation Trends in Computing and Communication*, 5(6), 1011-1016.

152. Djanali, S., Arunanto, F. X., Pratomo, B. A., Baihaqi, A., Studiawan, H. and Shiddiqi, A. M. (2014). Aggressive web application honeypot for exposing attacker's identity. IEEE In Information Technology, Computer and Electrical Engineering (ICITACEE), 2014 1st International Conference on, 212-216.

153. Yang, H. S. (2015). A study on attack information collection using virtualization technology. *Multimedia Tools and Applications*, 74(20), 8791-8799.

154. İnternet: Muscat, I. Cyber Threats vs Vulnerabilities vs Risks. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.acunetix.com%2Fblog%2Farticles%2Fcyber-threats-vulnerabilities-risks%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

155. Jabiyev, B. (2016*). Generating application layer IDS rules from cyber threat intelligence*, Doktora Tezi, İstanbul Şehir Üniversitesi, İstanbul.

156. İnternet: From character to personality. Trusted Automated eXchange of Indicator Information (TAXII™). URL: http://www.webcitation.org/query?url=https%3A%2F%2Foasis-open.github.io%2Fcti-documentation%2Ftaxii%2Fintro&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

157. Abu, M. S., Selamat, S. R., Ariffin, A. and Yusof, R. (2018). Cyber Threat Intelligence–Issue and Challenges. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(1), 371-379.

158. Fransen, F., Smulders, A. and Kerkdijk, R. (2015). Cyber security information exchange to gain insight into the effects of cyber threats and incidents. *E & I Elektrotechnik und Informationstechnik*, 132(2), 106-112.

159. İnternet: From character to personality. Structured Threat Information Expression (STIX™). URL: http://www.webcitation.org/query?url=https%3A%2F%2Foasis-open.github.io%2Fcti-documentation%2Fstix%2Fintro.html&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

160. İnternet: From character to personality. Cyber Observable eXpression (CybOX™). URL: http://www.webcitation.org/query?url=http%3A%2F%2Fcyboxproject.github.io%2Fdocumentation%2Fuse-cases&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

161. Appala, S., Cam-Winget, N., McGrew, D. and Verma, J. (2015). *An actionable threat intelligence system using a publish-subscribe communications model*. In Proceedings of the 2nd ACM Workshop on Information Sharing and Collaborative Security, 61-70.

162. İnternet: From character to personality. The VERIS Framework. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fveriscommunity.net%2Findex.html&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

163. İnternet: From character to personality. IBM X-Force Threat Intelligence. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.ibm.com%2Fsecurity%2Fxforce&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

164. İnternet: From character to personality. Introduction – User guide of MISP Malware Information Sharing Platform, a Threat Sharing Platform. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.circl.lu%2Fdoc%2Fmisp%2F&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

165. Debar, H., Becker, M. and Siboni, D. (1992). *A neural network component for an intrusion detection system*. In Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy, 240-250.

166. Debar, H., Dacier, M. and Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8), 805-822.

167. Lee, W., Stolfo, S. J., Chan, P. K., Eskin, E., Fan, W., Miller, M., Hershkop, S. and Zhang, J. (2001). *Real time data mining-based intrusion detection*. In Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01, 89-100.

168. Lee, W. and Stolfo, S. (1998). *Data mining approaches for intrusion detection*. Proceedings of the 7th USENIX Security Symposium.

169. Mavroeidis, V. and Bromander, S. (2017). *Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence*. In Intelligence and Security Informatics Conference (EISIC), 2017 European, 91-98.

170. Singh, S. and Gupta, P. (2014). Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27(27), 97-103.

171. Miles, B. J. and Kattan, M. W. (1995). Computer modeling of prostate cancer treatment: A paradigm for oncologic management?. *Surgical Oncology Clinics of North America*, 4(2), 361-372.

172. Sharafaldin, I., Lashkari, A. H. and Ghorbani, A. A. (2018, January). *Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, 108-116.

173. Panigrahi, R. and Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems. *International Journal of Engineering & Technology*, 7(3.24), 479-482.

174. Akhi, A. B., Kanon, E. J., Kabir, A. and Banu, A. (2019). *Network Intrusion Classification Employing Machine Learning: A Survey*, Doktora Tezi, Department of Computer Science and Engineering United International University

175. Eskin, E., Arnold, A., Prerau, M., Portnoy, L. and Stolfo, S. (2002). *A geometric framework for unsupervised anomaly detection*. In Applications of data mining in computer security, 77-101.

176. Aggarwal, P. and Sharma, S. K. (2015). Analysis of KDD dataset attributes-class wise for intrusion detection. *Procedia Computer Science*, 57, 842-851.

177. Jaiganesh, V., Mangayarkarasi, S. and Sumathi, P. (2014). An efficient algorithm for network intrusion detection system. *International Journal of Computer Applications*, 90(12).

178. Kumar, S. and Jain, S. (2012). Intrusion detection and classification using improved ID3 algorithm of data mining. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1, 352-356.

179. Araújo, N., de Oliveira, R., Shinoda, A. A. and Bhargava, B. (2010). *Identifying important characteristics in the KDD99 intrusion detection dataset by feature selection using a hybrid approach.* In 2010 17th International Conference on Telecommunications, 552-558.

180. Soni, P. and Sharma, P. (2014). An intrusion detection system based on KDD-99 data using data mining techniques and feature selection. *International Journal of Soft Computing and Engineering (IJSCE)*, 4(3).

181. Balogun, A. O. and Jimoh, R. G. (2015). Anomaly intrusion detection using an hybrid of decision tree and K-nearest neighbor. *Journal of Advances in Scientific Research & Applications (JASRA)*, 2(1), 67-74.

182. Elhafeez, E. A., Gody, A. M., Barakat, T. M. and Madbouly, A. I. (2017). Efficient Model for Intrusion Detection using Enhanced Classification Technique. *International Journal of Engineering Trends and Technology (IJETT)*, 97-104.

183. Sindhu, S. S. S., Geetha, S. and Kannan, A. (2012). Decision tree based light weight intrusion detection using a wrapper approach. *Expert Systems with Applications*, 39(1), 129-141.

184. Sutanto, D. H. and Ghani, M. K. A. (2015). Improving classification performance of K-nearest neighbour by hybrid clustering and feature selection for non-communicable disease prediction. *Engineering and Applied Sciences*, 10(16).

185. Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A. A. (2009). *A detailed analysis of the KDD CUP 99 data set*. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 1-6.

186. Rai, K., Devi, M. S. and Guleria, A. (2016). Decision tree based algorithm for intrusion detection. *International Journal of Advanced Networking and Applications*, 7(4), 2828.

187. Dhanabal, L. and Shantharajah, S. P. (2015). A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6), 446-452.

188. Mallissery, S., Kolekar, S. and Ganiga, R. (2013). *Accuracy Analysis of Machine Learning Algorithms for intrusion Detection System using NSL-KDD Dataset*. International Conference on Future Trends in Computing and Communication, 29-33.

189. Meena, G. and Choudhary, R. R. (2017). *A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA*. In 2017 International Conference on Computer, Communications and Electronics (Comptelix), 553-558.

190. Shrivas, A. K., Mishra, P. K. (2016, September). Intrusion Detection System for Classification of Attacks with Cross Validation. *International Journal of Engineering Science Invention*, 5(9), 21-24.

191. Debar, H., Dacier, M. and Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks*, 31(8), 805-822.

192. İnternet: From character to personality. Organizing your project. URL: http://www.webcitation.org/query?url=http%3A%2F%2Fexploreflask.com%2Fen%2Flatest%2Forganizing.html&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

193. Perona, I., Arbelaiz Gallego, O., Gurrutxaga, I., Martín, J. I., Muguerza Rivero, J. F. and Pérez, J. M. (2017). *Generation of the database gurekddcup*. Technical Report EHU-KAT-IK-02-16, University of the Basque Country (UPV/EHU).

194. Sahu, S. K.,Sarangi S., and Jena, K. S. (2014). *A Detail Analysis on Intrusion Detection Datasets*. In 2014 IEEE International Advance Computing Conference (IACC), 1348-1353.

195. İnternet: From character to personality. HoneyDrive. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fbruteforcelab.com%2Fhoneydrive&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

196. Barbato, G., Barini, E. M., Genta, G. and Levi, R. (2011). Features and performance of some outlier detection methods. *Journal of Applied Statistics*, 38(10), 2133-2149.

197. Shankar, R. and Sundararajan, M. (2015). Manufacturing quality improvement with data mining outlier approach against conventional quality measurements. *Indian Journal of Science and Technology*, 8(15).

198. İnternet: Han, J. and Kamber, M. Data Mining: Concepts and Techniques Slides for Textbook—Chapter 7 URL: http://www.webcitation.org/query?url=http%3A%2F%2Fbzupages.com%2Fattachments%2F20614d1288736162-chapter-3-data-preparation-data-mining-concepts-techniques-han-kamber-ppt&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

199. Chen, X., Cheng, J., Ye, J., Jin, Y., Li, X. and Zhang, F. (2018). Locating Station of One-Way Carsharing Based on Spatial Demand Characteristics. *Journal of Advanced Transportation*, 2018.

200. Bhattacharya, S. and Selvakumar, S. (2016). Multi-measure multi-weight ranking approach for the identification of the network features for the detection of DoS and Probe attacks. *The Computer Journal*, 59(6), 923-943.

201. Farid, D. M., Harbi, N. and Rahman, M. Z. (2010). Combining naive bayes and decision tree for adaptive intrusion detection. *International Journal of Network Security & Its Applications (IJNSA)*, 2(2), 12-25.

202. Panda, M. and Patra, M. R. (2008). *A comparative study of data mining algorithms for network intrusion detection*. In 2008 First International Conference on Emerging Trends in Engineering and Technology, 504-507.

203. Panda, M. and Patra, M. R. (2009, January). *Ensemble of classifiers for detecting network intrusion.* In Proceedings of the International Conference on Advances in Computing, Communication and Control, 510-515.

204. Wee, Y. Y., Cheah, W. P., Tan, S. C. and Wee, K. (2011). Causal discovery and reasoning for intrusion detection using bayesian network. *International Journal of Machine Learning and Computing*, 1(2), 185.

205. Chebrolu, S., Abraham, A. and Thomas, J. P. (2005). Feature deduction and ensemble design of intrusion detection systems. *Computers & Security*, 24(4), 295-307.

206. Shrivas, A. K., Singhai, S. K. and Hota, H. S. (2013). An efficient decision tree model for classification of attacks with feature selection. *International Journal of Computer Applications*, 84(14).

207. İnternet: Abhashkumar, A. and Michael, R. Implementing an Intrusion Detection System using a Decision Tree. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fpdfs.semanticscholar.org%2Fcae6%2F53e114a2e1f46c85455b7f15927ab2d8cb96.pdf&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

208. Petersen, R. (2015). *Data mining for network intrusion detection: A comparison of data mining algorithms and an analysis of relevant features for detecting cyber-attacks*. Yükek Lisans Tezi, Mid Sweden University, Östersund, Sweden.

209. Mulyadi, R. (2010). *Evaluating Classification Algorithms for Intrusion Detection Systems*, Yüksek Lisans Tezi, Management Information Systems in The Eller College of Management at The University of Arizona, Tucson, Arizona, USA.

210. Sriraghavan, R. G. and Lucchese, L. (2008). *Data processing and anomaly detection in web-based applications*. In Machine Learning for Signal Processing, 2008. MLSP 2008. IEEE Workshop on, 187-192.

211. Kruegel, C. and Vigna, G. (2003). *Anomaly detection of web-based attacks*. In Proceedings of the 10th ACM conference on Computer and communications security, 251-261.

212. İnternet: Positive Technologies. Web Application Attack Statistics 2017 In Review. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.ptsecurity.com%2Fupload%2Fcorporate%2Fww-en%2Fanalytics%2FWeb-application-attacks-2018-eng.pdf+&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

213. Early, J. P., Brodley, C. E. and Rosenberg, C. (2003). *Behavioral authentication of server flows*. In 19th Annual Computer Security Applications Conference, 2003, 46-55.

214. İnternet: From character to personality. What is PostgreSQL? Introduction, History, Features, Advantages. URL: http://www.webcitation.org/query?url=https%3A%2F%2Fwww.guru99.com%2Fintroduction-postgresql.html&date=2019-04-09, Son Erişim Tarihi: 09.04.2019

ÖZGEÇMİŞ

Kişisel Bilgiler



Soyadı, adı	:	ARIKAN, Süleyman Muhammed
Uyruğu	:	T.C.
Doğum tarihi ve yeri	:	1991 Ankara
Medeni hali	:	Evli
Telefon	:	-
Faks	:	-
e-posta	:	arikan.sm@gmail.com

Eğitim Derecesi	Okul/Program	Mezuniyet yılı
Yüksek lisans	Gazi Üniversitesi Adli Bilişim	Devam Ediyor
Lisans	Pamukkale Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü	2015
Lise	Şehit Vural Arıcı Anadolu Lisesi	2009

İş Deneyimi, Yıl	Çalıştığı Yer	Görev
2016 – devam ediyor	Türkiye Bilimsel ve Teknolojik Araştırma Kurumu Bilişim ve Bilgi Güvenliği İleri Teknolojiler Araştırma Merkezi Siber Güvenlik Enstitüsü	Araştırmacı

Yabancı Dili

İngilizce

Yayınlar

1. ARIKAN, S , BENZER, R . (2018). Bir Güvenlik Trendi: Bal Küpü *Acta INFOLOGICA***,** 2 (1), 1-11. DOI: 10.30801/acin.356815



GAZİLİ OLMAK AYRICALIKTIR…