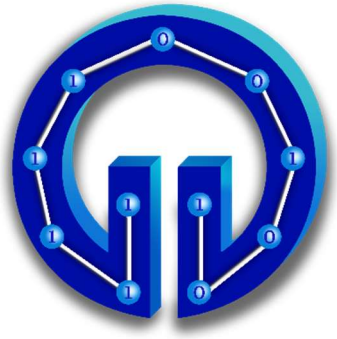
KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Ağ Saldırılarının Tespit Edilebilmesi İçin Makine Öğrenmesinin Kullanılması

BİTİRME PROJESİ

Ebrar KAYA
Kader GÜLTEKİN

2020-2021 GÜZ DÖNEMİ

KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Ağ Saldırılarının Tespit Edilebilmesi İçin Makine Öğrenmesinin Kullanılması

BİTİRME PROJESİ

Ebrar KAYA
Kader GÜLTEKİN

2020-2021 GÜZ DÖNEMİ

IEEE Etik Kuralları
IEEE Code of Ethics

Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriyi kabul etmek ve eleştiriyi yapmak; hatları kabul etmek ve düzeltmek, diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği veya cinsiyet ifadesi üzerinden ayırımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

Güvenli bir sistem kurmak için bilgi güvenliği risklerini azaltmak ve etkisiz kılmak, iş süreçlerinin sürekliliğini sağlamak, olası maddi ve manevi kayıpları minimize etmek, kurumsal prestiji korumak, güvenlik sorumluluklarının ve bilincinin artmasını sağlamak gerekmektedir.

Geleneksel saldırı tespit sistemlerine alternatif bir çözüm olarak makine öğrenmesi yöntemleriyle geliştirilen "Anomali Tabanlı Saldırı Tespit Sistemleri" kullanılması saldırı tespit sistemlerinin zayıflıklarına karşı ideal bir çözüm olarak karşımıza çıkmaktadır. Bu teknik yeni ağ saldırılarını tespit etmek için saldırı tespit sistemlerinin daha önceki saldırıları 'öğrenmesini! sağlar. Ayrıca gözetimsiz makine öğrenmesiyle desteklenen bir saldırı tespit sistemi ağın normal davranışını modelleyerek normalin dışındaki durumları yani anomalileri de tespit edebilir.

Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümünde Mühendislik Tasarım Dersi "Ağ Saldırılarının Tespit Edilebilmesi İçin Makine Öğrenmesinin Kullanılması" adlı bu çalışma ağ saldırılarını tespit etmek için gözetimli makine öğrenimi yaklaşımlarını kullanmayı amaçlamaktadır.

Üniversite hayatımız boyunca aldığımız çok yönlü eğitimin son demlerinde kariyerimize insanları, şirketleri, kurumları, organizasyonları ve hatta devletleri korumak için çalışan nice insanlardan olma fikriyle devam etme kararı aldık. Gelişmiş devletlerin bir özelliği olan iş bölümü ve uzmanlaşma kavramını benimseyerek, aldığımız eğitimi layıkıyla sonuçlandırmak, bir alanda uzmanlaşmak ve fayda sağlamak amacını taşıyoruz.

Öğrencileri için hiçbir özveriden kaçınmayan, mühendislik tasarım projemizin şekillenmesinde büyük emeği geçen, bizimle bilgi ve birikimlerini paylaşan, desteğini esirgemeyen danışman hocamız Doç. Dr. Güzin ULUTAŞ'a çok teşekkür ederiz. Ayrıca üzerimizde emeği olan tüm hocalarımıza, okul personelimize, arkadaşlarımıza ailelerimize teşekkür ederiz.

Ebrar KAYA
Kader GÜLTEKİN
Trabzon 2021

İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	1
ÖNSÖZ	2
İÇİNDEKİLER	3
ÖZET	4
1. GENEL BİLGİLER	6
1.1. Giriş	6
1.2. Motivasyon	7
1.3. Mühendislik Tasarım Projesi	7
1.4. Proje Tanımı	7
2. YAPILAN ÇALIŞMALAR	8
2.1. Makine Öğrenmesine Genel Bakış	8
2.1.1. Makine Öğrenimi Algoritmaları	9
2.1.2. Değerlendirme Ölçütleri	13
2.1.3. Ağ Yapısını Anlama	14
2.1.4. Ağ Üzerinde Gerçekleştirilen Saldırı Türleri	15
2.2. Proje İçin Yapılan Gereksinim Analizi ve Değerlendirmeler	16
2.3. Proje İçin Kullanılan Araçların Seçimi	17
3. ÇALIŞMANIN UYGULANMASI	18
3.1. Veri Ön İşleme	18
3.2. Öznitelik Seçimi (Feature Selection)	20
3.2.1. Filtreleme Yöntemleri	20
3.2.2. Sarmalayıcı Yöntemler (Wrapper Based)	20
3.2.3. Gömülü Metotlar (Embedded Methods)	21
3.3. Sınıflandırma (Classification)	22
3.3.1. Uygulanan Modeller	23
4. SONUÇ	26

5. KAYNAKLAR	31
STANDARTLAR ve KISITLAR FORMU	32

ÖZET

Her geçen gün internetin yaygınlaşması ve buna bağlı olarak ağa bağlanan cihazların hızlı bir şekilde artması, bazı avantajlarının yanında birçok sorunu da beraberinde getirmektedir. Bu sorunlardan en önemlisi siber tehditlerdir. Güvenli bir sistem kurmak için bilgi güvenliği risklerini azaltmak ve etkisiz kılmak, iş süreçlerinin sürekliliğini sağlamak, olası maddi ve manevi kayıpları minimize etmek, kurumsal prestiji korumak, güvenlik sorumluluklarının ve bilincinin artmasını sağlamak gerekmektedir. Kişilere, kurumlara ve devletlere karşı siber tehditler, maddi zararların yanında itibar ve zaman gibi zararlara da sebep olabilmektedirler.

Saldırı tespit ve saldırı önleme sistemleri, bu kayıpları ortadan kaldırmak veya en aza indirilebilmek için kullanılmaktadır. Saldırı tespit sistemleri imza tabanlı veya anomali tabanlı olarak tasarlanmakta ve günümüzde anomali tabanlı sistemler makine öğrenmesi yöntemleri kullanılarak geliştirilmektedir. Bu çalışmanın amacı, bir bilgisayar ağına saldırı olup olmadığını yüksek başarı oranı ile tespit etmektir. Bu sistemi geliştirmek için makine öğrenmesi yöntemlerinden Destek Vektör Makinesi (SVM- Support Vector Machine), Karar Ağacı (Decision Tree), Naive Bayes ve Random Forest kullanılmıştır. Sistemin geçerliliğini sınamak üzere "Intrusion Detection Evaluation Dataset (CIC-IDS2017)" veri seti kullanılmıştır.

Projemizin birinci bölümünde projenin tanımı ve proje ile ilgili genel açıklamalar yapılarak projeye genel bir bakış sağlanmıştır. İkinci bölümde ise makine öğrenmesine genel bakış, makine öğrenimi türleri ve algoritmaları, veri setinde kullanılan yaygın ağ saldırıları, performans değerlendirme ölçütleri ve proje için kullanılacak araçların seçimi anlatılmıştır. Üçüncü bölümde; makine öğrenmesi yöntemlerinin veri seti üzerinde gerçekleştirilmesi sağlanarak proje gerçeklenmiştir. Dördüncü bölümde ise; kullanılan modellerin herbirinin performans ölçütlerine göre değerlendirmesi yapılarak proje sonuçlandırılmıştır.

1. GENEL BİLGİLER

1.1. Giriş

Türk Dil Kurumu "güvenlik" kavramını; toplum yaşamında yasal düzenin aksamadan yürütülmesi, kişilerin korkusuzca yaşayabilmesi durumudur şeklinde açıklamıştır. Güvenlik, toplumun, onu oluşturan bireylerin, onların kişilik hakları ve insanlık onurlarının, kamusal ve kişisel malların, her türlü tehlike ve kazalardan korunması anlamına gelmektedir. Amaç toplumun refah ve huzur içerisinde varlığının korunması ve kamu düzeninin tesis edilmesidir. Güvenlik çok yönlü bir kavramdır. İnternetin hızla gelişmesi ile güvenliği tehdit edebilecek birçok yeni unsur eklenmiştir. Bu nedenle zaman içinde güvenlik kavramı siber güvenlik başlığı altında yeni bir alan daha kazanmıştır. Siber güvenlik bilginin korunmasıdır ve bilginin korunmasının; internet güvenliğinin, bilgisayar güvenliğinin, mobil güvenliğin ve ağ güvenliğinin sağlanması şeklinde dört yöntemi vardır.

Söz konusu özne kişiler, kurumlar veya organizasyonlar olabilir ancak nasıl ki askeri güvenlik önemli bir husussa bilgi güvenliği de o denli önemli bir husustur. Bilgi mahremdir ve gizliliğinin, bütünlüğünün ve erişilebilirliğinin sağlanması siber güvenliğin en temel amacıdır. Araştırmamızın amacı ağ üzerinden gerçekleştirilebilecek saldırıların makine öğrenmesi yöntemleri ile en yüksek başarı oranlarıyla tespit edilebilmesini sağlamaktır.

Bu araştırma Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü Mühendislik Tasarım Dersi kapsamında yapılmıştır. Çalışmamız ders öğretim üyesi Doç. Dr. Güzin ULUTAŞ gözetiminde yürütülmüştür. Çalışmamızın ana konusu "Ağ Saldırılarının Tespit Edilebilmesi İçin Makine Öğrenmesinin Kullanılması"dır.

1.2. Motivasyon

Güvenli bir sistem kurmak için bilgi güvenliği risklerini azaltmak ve etkisiz kılmak, iş süreçlerinin sürekliliğini sağlamak, olası maddi ve manevi kayıpları minimize etmek, kurumsal prestiji korumak, güvenlik sorumluluklarının ve bilincinin artmasını sağlamak gerekmektedir.

Ağ güvenliğini sağlamak için kullanılan birçok yöntemle beraber saldırı tespit sistemleri geçmişten günümüze önemli bir gelişim göstermiştir. Geleneksel saldırı tespit sistemleri önceden bilinen saldırıların tespit edilmesinde oldukça kullanışlı olmasına rağmen 0. Gün saldırılarına karşı savunmasızdırlar. Ek olarak yeni saldırı türleri ortaya çıktıkça saldırı tespit sistemlerinin imzalarının güncellenmesi gerekmekte ve bu ek iş yükü getirmektedir.

Geleneksel saldırı tespit sistemlerine alternatif bir çözüm olarak makine öğrenmesi yöntemleriyle geliştirilen "Anomali Tabanlı Saldırı Tespit Sistemleri" kullanılması saldırı tespit sistemlerinin zayıflıklarına karşı ideal bir çözüm olarak karşımıza çıkmaktadır. Bu teknik yeni ağ saldırılarını tespit etmek için saldırı tespit sistemlerinin daha önceki saldırıları 'öğrenmesini' sağlar. Ayrıca gözetimsiz makine öğrenmesiyle desteklenen bir saldırı tespit sistemi ağın normal davranışını modelleyerek normalin dışındaki durumları yani anomalileri de tespit edebilir.

Bu tez ağ saldırılarını tespit etmek için gözetimli makine öğrenimi yaklaşımlarını kullanmayı amaçlamaktadır.

1.3. Mühendislik Tasarım Projesi

Projemizin temeli olan Mühendislik Tasarım Projemizde çeşitli güvenlik duvarı teknolojilerinin ve saldırı tespit sistemlerinin araştırılması ve örneklendirilmesi, sistemlerin izlenme ve raporlanması gerçekleştirilmiştir.

İşletim sistemiyle gelen güvenlik önlemleri dışında hiçbir önlem almayan bir web sunucusu güvenlik duvarı, imza tabanlı saldırı tespit sistemi ve izleme ve raporlama sistemi kullanılarak savunulmuş ve ardından yapılan çalışmalar test edilmiştir.

Yapılan testler sonucunda ağı korumanın faydaları ve imza tabanlı saldırı tespit sisteminin yetersizlikleri ortaya konulmuştur.

1.4. Proje Tanımı

Bu tezin amacı makine öğrenimi yöntemlerinin saldırı tespit sistemlerinde kullanımının araştırılmasıdır. Bunu yapmak için farklı öğrenme algoritmaları araştırılmış, bir veri seti seçilmiş ve seçilen algoritmalar ile bu veri seti kullanılarak makine öğrenmesi gerçekleştirilmiştir.

2. YAPILAN ÇALIŞMALAR

2.1. Makine Öğrenmesine Genel Bakış

Makine öğrenimi, bilgisayarların algılayıcı verisi ya da veri tabanları gibi veri türlerine dayalı öğrenimini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini konu edinen bir bilim dalıdır. Makine öğrenimi araştırmalarının odaklandığı konu bilgisayarlara karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmaktır.

(https://tr.wikipedia.org/wiki/Makine_öğrenimi - Erişim Tarihi: 01.05.2021)

Bir makine öğrenimi projesi, model için girdilerin hazırlanması, modelin eğitilmesi ve son olarak modelin ne kadar iyi performans sergileyeceğini görmek için modelin test edilmesi aşamalarını içerir. Girdi (Input) genellikle probleme özel hazırlanmış ilgili veri setlerinden elde edilir. Özellikler (Features), analiz edilmesi istenilen nesneyi tanımlamak için kullanılan ölçümlerdir ve veri seti tarafından önceden tanımlanmış olabilecekleri gibi sonradan belirlenmeleri de gerekebilir (Feature extraction). Ancak seçilen özellikler amaçlanan makine öğrenmesi projesine uygun olmalıdır. Örneğin; ağ saldırılarının tespit edilmesi için kullanılacak bir makine öğrenimi projesinin veri seti daha önce gerçekleşmiş ağ saldırılarını içeriyorsa özellikleri de kaynak ip adresi, hedef ip adresi, kaynak portu, hedef portu, protokol bilgisi, paket boyutları gibi bu saldırıları tanımlayıcı yönde özellikler olmalıdır. Özellik vektörü (Feature vector) ise n adet özelliği içeren n-boyutlu bir dizidir. Bir model eğitileceği zaman bu özellik vektörü kullanılır. Model eğitildikten sonra ise çeşitli test verileriyle modele tahminler yaptırılarak doğruluk oranı sınanır ve bu doğrultuda modele iyileştirmeler yapılır.

Makine Öğrenimi Türleri

Farklı öğrenme türleri vardır. Gözetimli öğrenmede (Supervised learning), etiketlenmiş gözlemlerden öğrenme sürecidir. Etiketler, algoritmaya gözlemleri nasıl etiketlemesi gerektiğini öğretir, öyle ki model verilen özellik vektörlerine göre doğru bir etiketi nasıl tahmin edeceğini öğrenir. Örneğin aynı ip adresinden, x kadar sürede, y sayıda, z protokolünde paket geliyorsa bu bir q saldırısıdır denilebilir.

Sınıflandırma (Classification): Sınıflandırma, önceden tanımlanmış veri setlerindeki her bir gözleme bir kategori/sınıf ataması yapılmasıdır.
Regresyon (Regression): Eğitilen modelin her gözlem için öğrendiklerinden yola çıkarak reel bir değer tahmini yapmasıdır.

Bir modelin etiketsiz bir veri kümesi ile eğitilmesi ise gözetimsiz öğrenmedir (Unsupervised learning). Gözlemlerden öğrenme sürecidir. Algoritmanın kendi kendine keşifler yapması beklenir. Örneğin normal ağ davranışını modellemek ve anormal durumlar algılandığında alarm vermek için tasarlanacak bir makine öğrenmesi algoritmasında gözetimsiz öğrenme yöntemi kullanılır.

Kümeleme (Clustering): Gözlemleri, aynı küme içindeki nesnelerin diğer kümelerinkine kıyasla benzer özelliklere sahip olacağı şekilde homojen bölgelere ayırır.

Yarı gözetimli öğrenme (Semi-supervised learning), pekiştirmeli öğrenme (reinforcement learning) ve derin öğrenme (deep learning) ise diğer öğrenme yaklaşımlarıdır. Bu projede kullanılmadıkları için nasıl çalıştıklarıyla ilgili ayrıntılar atlanmıştır.

Hem sınıflandırma hem de kümeleme, bu proje için uygun yaklaşım türleridir. Sınıflandırma, saldırı türlerini tahmin etmeyi, kümeleme ise verileri normal ve anormal durumlar olarak gruplara ayırmayı sağlayacaktır.

Özellik Seçimi (Feature Selection)

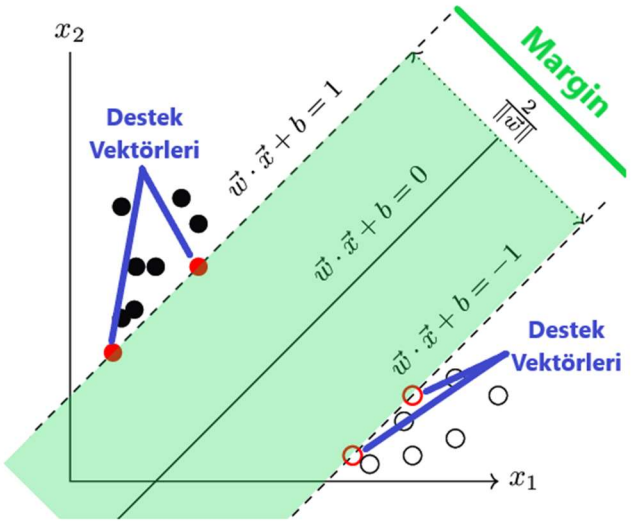
Herhangi bir makine öğrenimi modelinde özellik seçimi çok önemlidir. Dahil edilen özelliklerin modelin doğruluğunu artıran özellikler olması gerekir. Bu nedenle gözlemlerin mevcut özellik sayısı az ve öz olmalıdır. Özellik seçimi, özellik sayısını azaltmamıza olanak tanır ve bu da daha hızlı eğitim yapılmasına olanak tanır.

2.1.1. Makine Öğrenimi Algoritmaları

Aşağıda verilen algoritmalar bazı algoritmalara genel bakış niteliğindedir.

Destek Vektör Makinesi (Support Vector Machine)

Destek Vektör Makineleri (Support Vector Machine) genellikle sınıflandırma problemlerinde kullanılan gözetimli öğrenme yöntemlerinden biridir. Bir düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfının noktaları için de maksimum uzaklıkta olmasını amaçlar. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur.

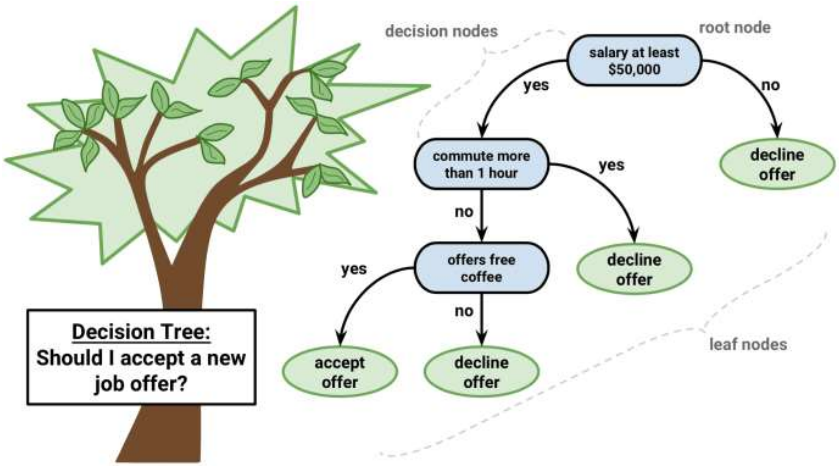


Şekil 1, Destek Vektör Makine Algoritması, medium.com/deep-learning-turkiye

Karar Ağacı (Decision Tree)

Karar Ağaçları (DT), hem sınıflandırma (Classification) hem de regresyon (Regression) problemlerinde kullanılan bir gözetimli öğrenme algoritmasıdır. Sınıflandırılmak istenen bir veriyi daha önceki verilerle olan yakınlık ilişkisine göre sınıflandıran bir algoritmadır. Amacı veri özelliklerinden basit kurallar çıkarıp bu kuralları öğrenerek bir değişkenin değerini tahmin eden bir model oluşturmaktır. Çok sınıflı verilerin kullanımında kullanılır. Sayısal veya sayısal olmayan değerler üzerinde işlemler yapar.

Bir ağaç yapısı, kök olarak adlandırılan bir tekil başlangıç düğümünden başlayarak dallanan düğümler topluluğu gibi karakterize edilebilir. Bir elemanın alt elemanı çocuk ve çocuğun üst elemanı da ebeveyn olarak adlandırılmaktadır. Aynı ebeveynin çocukları kardeş olarak adlandırılır. Çocuğun çocuğu olan düğüme torun, ebeveynin ebeveyni olan düğüme ise ata düğüm denir. Çocuğu olmayan düğümlere ise yaprak adı verilmektedir. Torunlara sahip düğümlerden oluşan bir ağaç içerisindeki her düğüm bir alt ağacın köküdür. Ebeveynden çocuklarına ve ondan da torunlarına tek bir yol boyunca hareket edilebilir. Bir kök ile bir düğüm arasındaki yol, bir düğümün seviyesi olarak adlandırılan ölçütü ortaya çıkarmaktadır. Derinlik, bir ağaç içerisindeki herhangi bir düğümün maksimum seviyesi veya ağacın katman sayısı olarak tanımlanabilir. Derece, düğümün alt düğümlerinde bulunan en çok eleman sayısıdır. (Prof. Dr. Vasif NABİYEV, Teoriden Uygulamalara Algoritmalar, 5. Basım, 2016, s. 94)



Şekil 2, Karar Ağacı Algoritması Örneği, medium.com/@ekrem.hatipoglu

Naive Bayes

Naive Bayes sınıflandırma algoritması, adını Matematikçi Thomas Bayes'den alan bir sınıflandırma/ kümeleme algoritmasıdır. Naive Bayes sınıflandırması olasılık ilkelerine göre tanımlanmış bir dizi hesaplama ile, sisteme sunulan verilerin sınıfını yani kümesini tespit etmeyi amaçlar.

Naive Bayes sınıflandırmasında sisteme belirli bir oranda öğretilmiş veri sunulur. Öğretim için sunulan verilerin mutlaka bir sınıfı/kümesi bulunmalıdır. Öğretilmiş veriler üzerinde yapılan olasılık işlemleri ile, sisteme sunulan yeni test verileri, daha önce elde edilmiş olasılık değerlerine göre işletilir ve verilen test verisinin hangi kümede olduğu tespit edilmeye çalışılır. Elbette öğretilmiş veri sayısı ne kadar çok ise, test verisinin gerçek kümesini tespit etmek o kadar kesin olabilmektedir. (https://kodedu.com/2014/05/naive-bayes-siniflandirma-algoritmasi/ Erişim Tarihi:01.05.2021)

K En Yakın Komşu (K Nearest Neighbour)

K en yakın komşuluk (KNN) algoritması, uygulaması gözetimli öğrenme algoritmalarındandır. Hem sınıflandırma hem de regresyon problemlerinin çözümünde kullanılıyor olmakla birlikte, endüstride çoğunlukla sınıflandırma problemlerinin çözümünde kullanılmaktadır.

Örnek veri setine katılacak olan yeni verinin, mevcut verilere göre uzaklığı hesaplanıp, k sayıda yakın komşuluğuna bakılır. Uzaklık hesapları için genelde 3 tip uzaklık fonksiyonu kullanılmaktadır:

- "Euclidean" Uzaklık
- "Manhattan" Uzaklık
- "Minkowski" Uzaklığı'dır.

K en yakın komşuluk (KNN) algoritması eğitim verilerini test zamanında kullanılmak üzere depoladığından eğitim süresine ihtiyaç duymaz ancak uzaklık hesabı yaparken bütün durumları depolayacağından, büyük veriler için kullanıldığında büyük miktarda bellek alanına gereksinim duyar.

Şekil 3'te KNN Modeli için doğruluk oranı %99 olarak görülmektedir.

Şekil 3, KNN Accuracy

K-Means Kümeleme (K Means Clustering)

Kümeleme (Clustering) bir veri setinde benzer özellikler gösteren verilerin gruplara ayrılmasına denir. Aynı küme içinde benzerlikler fazla, kümeler arası benzerlikler azdır.

K-Means kümeleme algoritması bir gözetimsiz öğrenme (unsupervised learning) ve kümeleme algoritmasıdır. K-Means' teki K değeri küme sayısını belirler ve bu değeri parametre olarak alması gerekir.

K değeri belirlendikten sonra algoritma rastgele K tane merkez noktası seçer. Her veri ile rastgele belirlenen merkez noktaları arasındaki uzaklığı hesaplayarak veriyi en yakın merkez noktasına göre bir kümeye atar. Daha sonra her küme için yeniden bir merkez noktası seçilir ve yeni merkez noktalarına göre kümeleme işlemi yapılır. Bu durum sistem kararlı hale gelene kadar devam eder.

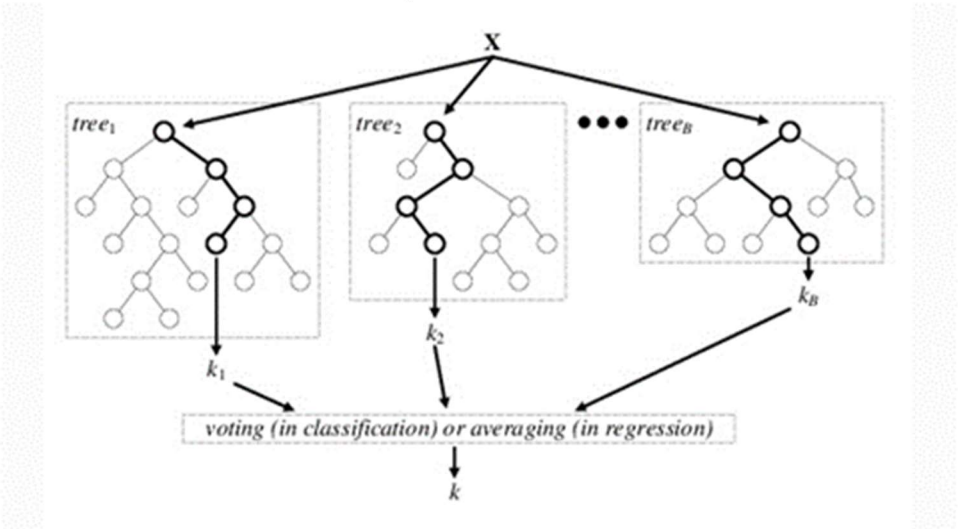
(https://medium.com/@ekrem.hatipoglu/machine-learning-clustering-kümeleme-k-means-algorithm-part-13-be33aeef4fc8, Erişim Tarihi: 30.05.2021)

K-means kümeleme algoritmasının, veri setini saldırı (attack) ve saldırı değil (benign) olarak iki kümeye ayırması beklenmektedir. Bu yöntem daha önce kullandığımız gözetimli öğrenme yöntemlerinin aksine bir gözetimsiz öğrenme yöntemidir, yani etiketli verileri kullanmaz, gözlemlerden öğrenir, algoritmanın kendi kendine keşifler yapması beklenir. Yani kesin bir doğru/yanlış sonucu üretmesi beklenmez bunun yerine gerçek zamanlı olarak tüm girdi verileri, öğrenilenlerin varlığında analiz edilir, etiketlenir ve kümelere yakınlığına göre bir sonuç üretilir.

Random Forest Algoritması

Rastgele orman algoritması, gözetimli bir makine öğrenmesi algoritmasıdır. Rastgele orman algoritması, hiperparatmetre ayarı yapılmadan iyi sonuçlar vermesi ve hem sınıflandırma hem regresyon problemlerine uygulanabilmesiyle en çok tercih edilen algoritmalardan biridir. Rastgele ormanın en büyük avantajı hem sınıflandırma hem regresyon problemlerinde kullanılabilmesidir.

Rastgele orman algoritmasının anlaşılabilmesi için Karar ağaçlarının anlaşılması gerekmektedir. Karar ağaçlarının en büyük problemlerinden biri aşırı öğrenmedir (overfitting). Rastgele orman algoritması bu problemin iyileştirilmesi için kullanılan bir algoritmadır. Problemin çözülebilmesi için hem veri setinden hem öznitelik setinden rastgele olmak üzere farklı setler seçilir bu setleri eğitilir. Bu yöntemle yüzlerce karar ağacı oluşturulur ve her bir karar ağacı kendi içinde tahminde bulunur. Son olarak eğer problemimiz bir regresyon ise karar ağaçlarından elde edilen tahminler toplanarak ortalaması alınır, eğer problemimiz bir sınıflandırma ise tahminler arasında en çok oy alan seçilir.



Şekil 4

```
print("Random Forest Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Random Forest Accuracy: 0.9989090497593545
```

Şekil 5, Random Forest Accuracy

2.1.2. Değerlendirme Ölçütleri (Evaluation Metrics)

Projede kullanılan farklı makine öğrenimi algoritmalarının kalitesini değerlendirmek için çeşitli metrikler kullanılabilir.

Hata matrisi, genellikle bir sınıflandırma modelinin gerçek değerlerin bilindiği bir dizi test verisi üzerindeki sınıflandırma algoritmasının performansını görselleştirmemizi sağlar. Bu tablo tahmin edilen değerlerin ve gerçek değerlerin dört farklı kombinasyonunu içerir.

	Predicted Attack	Predicted Benign
Truly Attack	True Positive (TP)	False Negative (FN)
Truly Benign	False Positive (FP)	True Negative (TN)

Şekil 6, Hata Matrisi (Confusion Matrix)

Doğruluk (Accuracy), doğru tahminlerin toplam tahmin sayısına oranıdır. Accuracy'nin tersi de misclassification rate'tir, bu da (false positive+false negative) / toplam tahmin sayısı olarak ifade edilebilir, hata oranını verir.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Şekil 7, Doğruluk (Accuracy)

Kesinlik (Precision), saldırı olduğu için saldırı olarak kabul edilenlerin, saldırı olmadığı halde saldırı olarak kabul edilenlerle saldırı olduğu için saldırı olarak kabul edilenlerin toplamına oranıdır. Eğer kesinlik (precision) oranı düşükse bu false-positive oranının yüksek olduğu manasına gelir. False-positive, saldırı olmadığı halde saldırı olarak tespit edilen verileri temsil etmektedir ve false-positive oranı ne kadar fazla ise sistemin kullanılabilirliği o kadar düşük olur.

$$Precision = \frac{TP}{TP + FP}$$

Şekil 8, Kesinlik (Precision)

Recall, doğru yapılan tespitlerin, yapılan saldırılara oranıdır. Yani saldırı olduğu için saldırı kabul edilenlerin, saldırı olduğu için saldırı kabul edilenlerle saldırı olduğu halde saldırı kabul edilmeyenlerin toplamına oranıdır. Düşük recall oranı sonuçlarda çok fazla false-negative olduğunu gösterir. False-negative var olan saldırıların tespit edilememesi durumudur.

$$Recall = \frac{TP}{TP + FN}$$

Şekil 9, Recall

F1-Score, precision ve recall değerlerinin harmonik ortalamasıdır. Hem precision hem de recall tek bir metrikte birleştirilerek, iki değer arasında bir denge sağlanır. Precision ve recall değerlerinin bu bağlamda dikkate alınması önemli olduğundan, modelin performansını değerlendirmek için F1 puanı kullanılır.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Şekil 10, F1 Score

2.1.3. Ağ Yapısını Anlama

Tezde ele alınan saldırı türlerinin tümü ağ saldırılarıdır. Bu nedenle bu bölümde, ağ ile ilgili yaygın protokollere ve genel ağ teorisine kısa bir genel bakış sunulması hedeflenmektedir.

TCP / IP modeli, her katmanın belirli bir hizmet sağladığı verilerin nasıl işleneceğini ve aktarılacağını belirlediği beş farklı katmandan oluşur. Her katman bir alt katmandan alınan verileri işler ve bir üst katmana iletir.

Uygulama Katmanı (Application Layer), uygulamaların devamlılığı için takip edilen prosedürler boyunca değiş-tokuş ettiği mesajların format ve anlamlarını belirleyen en üst katmandır.

İletim Katmanı (Transport Layer), bir bilgisayar üzerindeki uygulama programından bir diğerindeki uyulama programına iletim sağlar. Bir alıcının veri kabul edebileceği maksimum hızı kontrol eden spesifikasyonlar, ağ tıkanmasını engelleyen mekanizmalar ve tüm verilerin doğru sırada ulaştığını garanti eden teknikler bu katmanda yer alır.

İnternet Katmanı (Network Layer), İnternet adresleme yapısı, İnternet protokollerinin formatı, geniş bir internet paketinin iletim için küçük paketlere bölünmesi yöntemleri ve hataları raporlama mekanizmalarının bulunduğu katmandır.
Ağ Arayüzü Katmanı (Network Interface Layer), ağ adresleri ve ağın destekleyebileceği maksimum paket boyutu hakkındaki spesifikasyonlar, adı geçen ortama erişim için kullanılan protokoller ve donanım adresleme gibi faaliyetlerin gerçekleştiği katmandır.

Fiziksel Katman (Physical Layer) ise iletim ortamı ve kullanılan donanım ile ilgili detayların belirlendiği en alt katmandır.

TCP/IP Katmanı	**Kullanılan Protokoller**
Uygulama Katmanı (Application Layer)	FTP, HTTP, SMTP
İletim Katmanı (Transport Layer)	TCP, UDP
İnternet Katmanı (Network Layer)	IP, ICMP
Ağ Arayüzü Katmanı (Network Interface Layer)	Ethernet, ARP

Tablo 1

Bu tezde incelenen saldırılar daha çok uygulama ve taşıma katmanında meydana gelmektedir.

İletim katmanı protokollerinden en çok bilinen iki tanesi TCP (Transmission Control Protocol) protokolü ve UDP (User Datagram Protocol) protokolleridir. TCP protokolü, istemci ile sunucu arasında bir bağlantı kurarak paketlerin kayıpsız teslim edilmesini sağlar. HTTP, HTTPS, POP3, SSH, SMTP, Telnet ve FTP gibi birçok protokol TCP iletişimi vasıtasıyla sağlanır. UDP protokolü, iletişim öncesi bağlantı kurmayı gerektirmediğinden daha az güvenilir bir İletim katmanı protokolüdür. TFTP, DNS ve SNMP gibi birçok protokol UDP iletişimi vasıtasıyla sağlanır.

Uygulama katmanındaki protokoller kullanıcıların etkileşimde bulunmasını sağlar. Örneğin; Telnet ve SSH (Secure Shell) uzak bir terminal sağlar, FTP (File Transfer) ve TFTP (Trivial File Transfer Protocol) dosya aktarımı sağlar ve SMTP (Simple Mail Transfer Protocol) e-posta desteği sunar.

2.1.4. Ağ Üzerinde Gerçekleştirilen Saldırı Türleri

Bu bölümde günümüzde yaygın olarak gerçekleştirilen farklı ağ saldırı türlerine genel bir bakış sunulacaktır.

Botnet

Botnet sözcüğü, "robot" ve "network (ağ)" sözcüklerinin birleşiminden türetilmiştir. Kötü niyetli görevleri gerçekleştirmek için birbirine bağlanan ve düşman tarafından kontrol edilen, güvenliği ihlal edilmiş cihazların (botlar) bir koleksiyonu olarak tanımlanabilir. DDoS saldırıları gerçekleştirmek, spam e-postalar göndermek, kötü amaçlı yazılım dağıtmak ve diğer birçok siber suçu gerçekleştirmek için kullanılırlar.

Brute Force

Brute force (kaba kuvvet) saldırısı, saldırganın bir hesaba yetkisiz erişim sağlayabilmek için deneme-yanılma yöntemi kullanmasına denir.

Denial of Service (DoS)

Hizmet dışı bırakma saldırıları, hedef sisteme çok fazla istek göndererek sistemin hizmet dışı kalmasını sağlayan saldırı tekniğidir. DoS saldırılarında bir makineden tek bir web sitesine veya internet servisine şişirilmiş istekler gönderilir. Web sitesi veya internet sitesi bu isteklerin tümüne yanıt veremez ve hizmet dışı bırakılmış olur.

Distributed Denial of Service (DDoS)

DoS saldırılarıyla aynı mantıkla yapılmaktadır. Aralarındaki önemli fark, saldırının çıkış kaynağıdır. DoS saldırılarında tek bir sunucudan hedefe doğru saldırı yapılır. DDoS saldırılarında ise şiddetinin arttırılması amacıyla tek bir sunucudan değil, çok sayıda sunucudan tek hedefe saldırı yapılmaktadır. Kullanıcısından habersiz ele geçirilmiş uzaktan yönetilebilen bilgisayarlara zombi bilgisayar denir ve bu saldırılar yapılırken zombi bilgisayarlardan oluşan botnet'ler kullanılabilmektedir. Doğru konfigüre edilmeyen ağ cihazları bu saldırıların başarısını artırmaktadır.

Web Saldırıları

Web saldırıları için kullanılan güvenlik açıkları genellikle uygulama katmanında bulunur. Uzaktan kod yürütme olarak da adlandırılan "Code injection" saldırıları, bir saldırganın uygulamaya kötü amaçlı kod enjekte etmesi ve yürütmesi şeklinde gerçekleşen bir saldırıdır. Saldırganın bir uygulamaya kod "enjekte etmesi" sahip olmaması gereken bilgilere veya daha kötüsü sistemin kendisine erişim elde etmesine yol açabilir. OWASP (The Open Web Application Security Project), "Code injection" saldırısını en riskli web uygulaması güvenlik riski olarak değerlendirmektedir. Siteler arası betik çalıştırma olarak tanımlayabileceğimiz "Cross site scripting (XSS)" ise, HTML kodlarının arasına istemci tabanlı kod gömülmesi yoluyla kullanıcının tarayıcısında istenen istemci tabanlı kodun çalıştırılabilmesi olarak tanımlanır. OWASP'ın listesinden en riskli web güvenlik açıkları listesinde ilk ondadır. (https://owasp.org/www-project-top-ten/, Erişim Tarihi: 30.05.2021)

2.2. Proje İçin Yapılan Gereksinim Analizi ve Değerlendirmeler

Projemiz için belirlediğimiz hedeflediğimiz öncelikli ve makine öğrenmesi için önemli gördüğümüz adımlar şu şekildedir;

- Projemize uygun veri setini belirleyip bu veri seti üzerinde en doğru sonucu verecek algoritmanın belirlenmesi için birden fazla makine öğrenmesi algoritmasının uygulanması.
- Proje tamamlandığında makine öğrenmesi algoritmalarıyla ulaşılacak doğruluğun optimum değerde olması.

Projenin planlanması ve takibinin yapılabilmesi için proje için belirlediğimiz görevler;

- Veri setinin belirlenmesi
- Veri setinin ön işlemesi
- Veri seti için Feature Selection
- Veri seti üzerinde makine öğrenmesi algoritmalarının uygulanması
- Modeller arasında karşılaştırmaların yapılması
- Veri setinden ayırdığımız test setiyle makine öğrenmesiyle yapılan tahminin karşılaştırılması

Şeklindedir.

2.3. Proje İçin Kullanılan Araçların Seçimi

Programlama Dili

Makine öğrenmesinde, Python, R, C++, C, C++, C, JavaScript gibi farklı programlama dilleri kullanılmaktadır. Yaptığımız literatür taramalarında birçok projede Python programlama dilinin kullanıldığını gördük. Python, içerdiği gelişmiş kütüphanelerle, söz dizimlerinin birbirinden farklı birçok makine öğrenmesi algoritmalarında uygulanabilirliğiyle, diğer dillere göre kullanım rahatlığıyla bize avantaj sağlayacağı için projemizde Python programlama dilini kullanmayı tercih ettik. Python'u tercih etmemizin bir diğer sebebi ise birçok projede kullanıldığından dolayı bize daha fazla kaynağa erişme avantajını sağlamasıydı.

Araçlar

Projemizi geliştirirken makine öğrenmesi için Jupyter Notebook uygulamasını kullandık. Jupyter Notebook, web tarayıcısı üzerinden kod çalıştırmamızı, düzenlememizi sağlayan server-client tabanlı bir uygulamadır. Kod yazarken program hücrelere bölünebilir, böylelikle yazılan kodlar hücre hücre çalıştırılabilir. Bu durum yaptığımız hataları, her adımda, kullandığımız veri setindeki değişiklikleri, durumları ve tabloları görebilmemizi ve analiz edebilmemizi sağlar.

Kütüphaneler

Python' da makine öğrenmesinde kullanabileceğimiz birçok hazır kütüphane bulunmaktadır. Projemizde kullandığımız veri setinin, veri ön işleme aşaması için NumPy ve Pandas kütüphanelerini kullandık. NumPy, çok boyutlu diziler ve matrisler üzerinde matematiksel hesaplamalar yapabileceğimiz bir Python kütüphanesidir. Büyük veriler üzerinde daha verimli ve daha az kodla çalışır. Pandas, kullanımı kolay, performansı yüksek olan veri

analiz araçları ve veri yapıları sağlayan bir Python kütüphanesidir. Pandas, data frame yapısını sağlamasıyla CSV dosyaları üzerinde işlemler yapılmasına olanak sağlar. Makine öğrenmesinde veri ön işleme ve veri okuma aşamalarında birçok kolaylık sağlar. Projemizin öğrenme aşamasında NumPy ve Pandas kütüphaneleriyle birlikte çalışan Scikit-learn kütüphanesini kullandık. Scikit-learn, denetimli ve denetimsiz birçok öğrenme algoritması sağlar. Kullanımı kolaydır ve bir makine öğrenmesi projesi için gerekli birçok aracı kullanıma sunar.

Veri Seti

Makine öğrenmesinde, öğrenme algoritmasına girdi olarak bir veri seti verilmesi gerekir. Makine öğrenmesinin doğru ve güvenilir sonuçlar vermesi için birbirinden farklı ve güncel saldırı türlerini içeren veri seti kullanmamız gerekir. Bunun için saldırı tespit sistemleri için hazırlanmış CICIDS2017 veri setini kullanmayı tercih ettik. CICIDS2017 veri seti PCAP' lere benzeyen zararsız, güncel saldırıları ve aynı zamanda ağ trafiği analizi sonuçlarını da içermektedir. Diğer veri kümelerinin kullanımıyla çıkan sorunları çözmeyi amaçlaması da bu veri setini seçmemizin nedenlerinden bir tanesidir.

BÖLÜM 3

ÇALIŞMANIN UYGULANMASI

3.1. Preprocessing (Veri Ön İşleme)

Makine öğrenmesinin ilk aşaması verinin kullanıma hazırlanması yani ön işleme kısmıdır. Veri ön işleme, makine öğrenmesinin sonuca etkisinin çok fazla olduğu en önemli aşamalardan biridir. Bunu bir binanın inşası olarak düşünebiliriz. Binanın temeli ne kadar doğru atılırsa ortaya çıkacak sonuç o kadar sağlam olacaktır. Veri ön işlemesini de makine öğrenmesinde temel atmak olarak görebiliriz.

Veri ön işlemede, seçtiğimiz veri seti üzerinde bazı işlemler yapıyoruz. Bu işlemleri; düzeltme, eksik veriyi tamamlama, tekrarlanan verileri kaldırma, dönüştürme, bütünleştirme, temizleme, normalleştirme, boyut indirgeme vb. olarak sıralayabiliriz.

Veri Setinin İçe Aktarılması

Veri setlerinin kullanılabilmesi için kullandığımız Jupyter Notebook içerisine aktarmamız gerekir. Veri setleri çoğunlukla *.csv* formatında bulunmaktadırlar. CSV formatındaki dosyalar verileri sütunlarda toplamak yerine virgüller ile ayırarak belli bir düzende yazıp depolar. Bu dosyalarda her satır bir veri kaydını temsil etmektedir. Bir CSV dosyasını içe aktarabilmek için Pandas kütüphanesi ile birlikte gelen *read_csv* metodu kullanılmaktadır. Projemizde kullandığımız veri seti birden fazla CSV dosyası içermektedir. Bu CSV dosyalarını listeleyebilmek için *glob* kütüphanesini kullandık. Dosyaların projede kullanılabilmesi için tek bir dosya olacak şekilde birleştirilmesi gerekmektedir. Glob kütüphanesi kullanılarak listelenen bu dosyaların birleştirilmesi için Pandas kütüphanesi ile birlikte gelen *pd.concat* metodu kullanılmaktadır.

Eksik Verilerin Ele Alınması

Veri setinin kullanıma hazırlanmasında ilk aşamalardan biri eksik verilerin bulunup bulunmadığının kontrolüdür. Eksik verilerin oluşmasına; yazılımsal, donanımsal hatalar, yanlış kayıtlar, veri kaynağında yaşanan bozulmalar gibi hatalar neden olabilmektedir. Eksik değerlerin ele alınması oluşturacağımız makine öğrenmesi modelinin sonucunun doğruluğunu iyileştirmek için uygulanması gereken aşamalardan biridir.

Eksik verilerin giderilmesi için kullanılan bazı algoritmalar vardır. Regresyon analizi, Hot deck, Cold deck, Naive Bayes gibi yöntemler bu algoritmalardan bazılarıdır. Biz projemizin bu kısmında eksik verilerin bulunduğu satırları veri setinden çıkardık. Veri seti oldukça büyük olduğundan dolayı eksik verilerin bulunduğu satırların veri setinden çıkartılması oluşturacağımız makine algoritması modelinin vereceği sonuca olumsuz bir etkisi olmayacaktır. Eksik verileri veri setinden çıkarırken ilk olarak eksik olan verilerin değerlerine NaN değeri atamasını yaptık. Daha sonra Pandas kütüphanesi ile birlikte gelen *dropna()* modülü ile *NaN* değerlerinin bulunduğu satırlar veri setinden çıkartılır.

Verilerin Eğitim ve Test Olarak Bölünmesi

Makine öğrenmesinde, özellikle denetimli öğrenmede model verilerle eğitilir. Kullanılan veri setindeki verilerin bir kısmını modeli eğitmek için, bir kısmını ise eğitilen modelin başarısını test etmek için bölmemiz gerekir. Eğitim ve test kümeleri için bazı oranlar vardır. Biz projemizde veri setinin %0 'lik kısmını test, %80 lik kısmını eğitim olacak şekilde iki parçaya böldük. Eğer veri setinin eğitim kısmını %90 'dan fazla olacak şekilde bölersek modelimiz bir süre sonra ezber yapar ve başarısı düşer. Bu gibi durumların önlenmesi için veri setinin nasıl bölüneceğine iyi karar vermek gerekir. Veri setinin bölünmesi için *sklearn.model_selection* kütüphanesi ile birlikte gelen *train_test_split* modülü kullanılabilir.

Özellik Ölçekleme (Feature Scaling)

Özellik ölçekleme, veri seti içerisindeki değerleri herhangi bir makine öğrenmesi algoritmasına sokmadan önce belirlediğimiz aralıklara çekilmesidir. Mesela bir veri setinde boy ve yaş değişkenleri olsun. Veri setindeki boy değişkenleri 130 ve 190 arasında değerler alırken, yaş değişkenleri 8 ve 65 arasında değerler alsın. Makine öğrenmesiyle oluşturduğumuz model bu değişkenler arasından boy değişkeninin yaş değişkeninden daha önemli olduğuna karar verir. Bu durumun oluşmaması için değişkenlerin hepsinin aynı aralığa çekilmesi gerekir. Standart Ölçekleme ve Min-Max Normalizasyon gibi yöntemler en çok kullanılan yöntemlerdir. Projemizde Min-Max Normalizasyon yöntemini tercih ettik.

* **Standart Ölçekleme**

Standart Ölçekleme yönteminde, değişkenlerin ortalaması 0, standart sapması 1 olan bir dağılama dönüştürülür. Teknik olarak, veri setindeki tüm verilerin ortalaması alınarak standart sapmalarına bölünerek normalleştirilmesidir. Bu yöntem ile veri setindeki değerler -1 ile 1 arasındaki değerlere çekilir. Elde edilen değere z-puanı (standart puan) denir. Formülü Şekil 11'de gösterildiği gibidir.

$$z = \frac{\chi - \mu}{\sigma}$$

Şekil 11, Standart Scaler Formülü

* **Min-Max Normalizasyon**

Min-Max Normalizasyonunda, her bir değer ilgili sütundaki minimum maximum değerlere göre ölçeklendirilir. Veri setindeki değerler 0 ile 1 arasındaki değerlere çekilir. Formülü Şekil 12'de verildiği gibidir.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Şekil 12, Min-Max Normalizasyon Formülü

3.2. Öznitelik Seçimi (Feature Selection)

Öznitelik seçimi, veri seti içerisinde bulunan verilerden ilgili özellikleri tespit etmek, ilgisiz veya az ilgili olan verileri kaldırmak için yapılmaktadır. Bu işlem makine öğrenmesi modelinin performansını önemli ölçüde etkilemektedir. Gereksiz özniteliklerin veri setinden çıkarılması; eğitim süresinin azalması, modelin doğruluğunun artması, overfitting durumunun engellenmesi gibi durumlarda oldukça fayda sağlamaktadır. Öznitelik seçimi yöntemlerinin büyük kısmı üçe ayrılmaktadır.

3.2.1 Filtreleme Yöntemleri

Özniteliklerin ilgi değerlerinin hesaplanabilmesi için bağımlı ve bağımsız değişkenler arasındaki ilişkinin karşılaştırıldığı yöntemdir. Filtreleme yöntemlerinde ilişki değerleri hesaplandıktan sonra veri setinde filtreleme yapılarak seçilen özelliklerle bir alt küme oluşturulur. Filtreleme yöntemlerinde Pearson Korelasyonu ve Ki- Kare en çok tercih edilenlerdendir.

Pearson Korelasyonu (Pearson Correlation)

Pearson korelasyonunda, doğrusal olarak birbiriyle ilişkili iki değişken arasındaki ilişkinin derecesinin ölçülmesinde kullanılır. Eğer veri setinde bulunan değişkenler sayısal ise bu yöntemin kullanılması daha doğru olur. Korelasyon katsayısı matematiksel olarak –1 ile 1 arasında değerler almaktadır. 1 'e yakın değerler pozitif ilişkinin olduğunu, -1 'e yakın değerler negatif ilişkinin olduğunu ifade etmektedir. 0 veya 0 'a yaklaşan değerler ilişkinin olmadığını belirtir.

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Şekil 13, Pearson Korelasyon Katsayısı

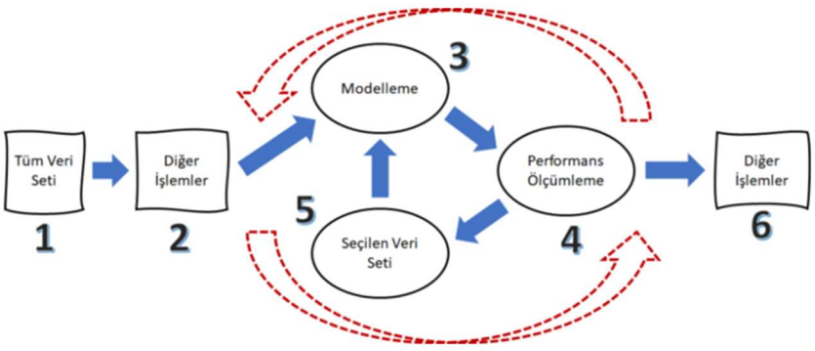
Ki Kare (Chi Square)

Ki Kare yöntemi sadece kategorik değişkenler için kullanılan bir yöntemdir. İki kategorik veri arasında ilişki olup olmadığını tespit ederek ilişkisi bulunmayan verilerin veri setinden çıkarılmasını sağlar. İki veri arasında ilişki bulunup bulunmadığı p değeri ile belirlenebilir.

Eğer p değeri <0,05 ise öznitelik hedef değişken için bağımlı, >=0,05 ise bağımsız bir değişkendir. Belirlenen öznitelik hedef değişken için bağımsız olan değişkenler veri setinden çıkarılır. *Sklearn.feature_selection* kütüphanesi ile birlikte gelen *SelectKBest* fonksiyonu bağımlı değişkenlerin seçilmesine olanak tanır. Fonksiyonda bulunan k değeri seçilecek en iyi özelliklerin sayısını belirtmektedir.

3.2.2 Sarmalayıcı Yöntemler (Wrapper Based)

Sarmalayıcı yöntemler, veri setindeki bağımsız değişkenlerden yola çıkarak oluşturulan alt kümeler ile bir çok model oluşturur. Bu modeller arasından en iyi performans gösteren modeli, seçilen alt küme değişkenleriyle belirler. Örnek olarak 100 bağımsız ve 1 bağımlı değişkenimiz olduğunu düşünelim. Sarmalayıcı yöntemlerden birini kullanarak 100 bağımsız değişken için birbirinden farklı değişken setleri oluşturulur. Her değişken kümesi kombinasyonu için model çalıştırılır. Bu yöntem zaman ve maliyet açısından filtreleme yöntemlerine göre daha avantajsızdır.



Şekil 14, Sarmalayıcı Yöntemler

İleri Doğru Seçim (Forward)

Modelin performanısına iyi anlamda en çok etki eden değişken seçilir. İlk değişken seçildikten sonra ikince değişkende aynı yöntemle seçilir her seferinde bu işlem devam eder. Bu döngü belirlenen performans ölçüsünün sınırına geldiğinde sonlanır.

Geriye Doğru Seçim (Backward)

Geriye doğru seçim ileri doğru seçimin tam tersi şeklinde işler. Modelin performansına bir katkısı olmayan veya kötü anlamda etki eden değişken seçilir. P değeri 0.05' e eşit veya yüksek olan değişkenler model veri setinden çıkartılır. Bu döngü veri setinde p değeri 0.05' ten küçük değişkenler kalana kadar devam eder.

3.2.3 Gömülü Metotlar (Embedded)

Gömülü metotlarda değişken seçimini modelin kendisi yapar. Hem sınıflandırma hem regresyon problemlerinde gömülü metotlarda bulunan algoritmalar kullanılabilir. Bağımsız değişkenlerin modele olan etkisini artırarak veya azaltarak modelin değişken yapısını ortaya çıkarır. Bu sayede hangi değişkenlerin modele daha az etki ettiğini gösterir ve değişkenler otomatik olarak seçilir. Sınıflandırma problemleri için karar ağacı tabanlı algoritmalar kullanılırken regresyon problemleri için düzenleme (regularization) yöntemleri kullanılmaktadır.

Lasso

Lasso, en güçlü düzenleme (regularization) yöntemlerindendir. Bağımsız değişkenlerin katsayılarını 0' a indirir ve katsayısı 0'a yaklaşan değişkenlerin modele etkisi olmadığı belirlenmiş olur. Şekil 15'de verilen formüldeki cost ve lambdanın değerleri algoritmanın performansını önemli ölçüde etkilemektedir.

$$\min_{\theta} \frac{1}{2} \sum_{i} (y_i - X_i\theta)^2 + \lambda ||\theta||_1$$

Şekil 15, Lasso Lineer Regression Formülü

Rigde

Lasso'dan farklı olarak değişkenleri 0'a indirmeye çalışmamaktadır bu yüzden kesin olarak seçilmek istenen değişkenler için kullanılmamaktadır. Bağımlı değişkeni en çok niteleyen bağımsız değişkenleri bulmak konusunda Lasso yöntemne göre daha başarılıdır.

Elastic Nets

Elastic Nets, Lasso Rigde yöntemlerinin birleşiminden oluşmaktadır. Lasso yönteminde kullanılan değişkenleri 0' a indirme işlemiyle, Rigde yönteminde kullanılan bağımlı değişkeni en çok niteleyen bağımsız değişkenleri bulma işlemini birleştirerek kullanır.

3.3. Sınıflandırma (Classification)

Makine öğrenmesinin ana konularından biri olan Gözetimli Öğrenme (Supervised Learning) iki ana başlığa ayrılır: Sınıflandırma (Classification) ve Regresyon (Regression). Sınıflandırma ve Regresyon, girdi verisi ile alakalı değildir, beklenen çıktı ile alakalıdır. Yani tezde ele alınan konudan yola çıkılırsa model eğitildikten sonra artık onun saldırıların türünü belirlemesi beklendiğinden kullanılacak yöntem sınıflandırma olacaktır ancak modelin x

saldırısının gelme ihtimalini hesaplaması bekleniyorsa bu regresyon olacaktır. Bu bağlamda kullanılacak yöntem sınıflandırma olacaktır.

Sınıflandırma, benzer özellikteki nesnelerin önceden belirlenmiş alt gruplara atanması işlemidir. Veriyi, veri kümesi üzerinde tanımlı olan çeşitli sınıflar arasında dağıtır. Sınıflandırma algoritmaları, verilen eğitim kümesinden bu dağılım şeklini öğrenirler ve daha sonra sınıfının belirli olmadığı test verileri geldiğinde doğru şekilde sınıflandırma çalışırlar.

Sınıflandırma yöntemlerine örnek olarak; Karar Ağaçları (Decision Trees), Naive Bayes, K En Yakın Komşuluğu (K Nearest Neighbours), Genetik Algoritmalar, Yapay Sinir Ağları, Rastgele Orman (Random Forest), Destek Vektör Makineleri (Support Vector Machines) verilebilir.

3.3.1. Uygulanan Modeller (Implemented Models)

Destek Vektör Makinesi (SVM- Support Vector Machine)

SVM'ler sınıflandırma işlemini yerine getiren bir gözetimli (supervised) öğrenme yöntemidir. Değişkenler arasındaki örüntülerin bilinmediği veri setlerindeki sınıflandırma problemleri için önerilmiş bir yöntemdir. İstatiksel öğrenme teorisine ve yapısal risk minimizasyonuna dayanmaktadır. Sınıflandırma için kullanılan basit ve etkili bir yöntemdir. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır, bu durum SVM'nin sınırı nasıl çizeceğini belirler. Örnek olarak el yazısı tanıma, yüz algılama, yaya algılama ve metin kategorizasyonu gibi örüntü tanıma problemlerini çözmek için kullanılabilir.

SVM'ler hiçbir parametre almayan (nonparametric) sınıflayıcılardır. Eğitim setlerinde girdiler ve çıktılar eşlenir. Eşler aracılığıyla test verilerinde ve yeni veri setlerinde girdi değişkeninin sınıflayacak karar fonksiyonları elde edilir. Hem doğrusal verilere hem de doğrusal olmayan verilere uygulanabilir. Çok sayıda bağımsız değişkene uygulanabilir ve karmaşık karar sınırlarını modelleyebilir. SVM'lerde overfitting problemi yoktur.

(Md Nasimuzzaman Chowdhury and Ken Ferens, Mike Ferens, Network Intrusion Detection Using Machine Learning)

Bu projede SVM kullanılarak düzlem "saldırı (attack)" ve "saldırı değil (benign)" olarak ikiye bölünmüştür. Doğruluk oranı %92 olarak belirlenmiştir.

```
In [24]: print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
         Accuracy: 0.9280750951242627
```

Şekil 16, SVM Accuracy

Karar Ağacı (Decision Tree)

Karar Ağaçları (DT), hem sınıflandırma (Classification) hem de regresyon (Regression) problemlerinde kullanılan bir gözetimli öğrenme algoritmasıdır. Sınıflandırılmak istenen bir veriyi daha önceki verilerle olan yakınlık ilişkisine göre sınıflandıran bir algoritmadır. Amacı veri özelliklerinden basit kurallar çıkarıp bu kuralları öğrenerek bir değişkenin değerini tahmin

eden bir model oluşturmaktır. Çok sınıflı verilerin kullanımında kullanılır. Sayısal veya sayısal olmayan değerler üzerinde işlemler yapar. Karmaşık veri setlerinde kullanımı kolaydır.

Karar ağaçlarının ilk hücrelerine kök (root veya root node) denir. Kök hücrelerinin altında düğümler (interval nodes veya nodes) bulunur. Her bir gözlem düğümler yardımıyla sınıflandırılır. Düğüm sayısı arttıkça modelin karmaşıklığı da artar. Karar ağacının en altında yapraklar (leaf nodes veya leaves) bulunur. Yapraklar, sonucu verir.

Kök hücreyi seçerken veri setini mümkün olduğunca anlamlı şekilde ayrıştırabilen sütun seçilmeye çalışır. Gini: Alt kümenin saflık değerinin formülü Şekil 17'de görülebilir.

$$Gini = 1 - \sum_j p_j^2$$

Şekil 17, Gini: Alt kümenin saflık değeri formülü

pj, j sınıfının gerçekleşme olasılığıdır. Her sınıf için hesaplanır ve çıkan sonuçların karelerinin toplamı birden çıkartılır. Gini değeri 0 ile 1 arasında bir sonuç alır ve sonuç 0'a ne kadar yakınsa o kadar iyi ayrım yapmış olur. (https://medium.com/deep-learning-turkiye/karar-ağaçları-makine-öğrenmesi-serisi-3-a03f3ff00ba5, Erişim Tarihi: 01.06.2021)

Kök (root) hücrenin bulunabilmesi için her düğüm için Gini değeri hesaplanır. Bu değerlerden sıfıra en yakın olanı kök olarak seçilir. Kökten sonra gelecek olan düğümlere de yine aynı yöntemle karar verilir. Projede Karar Ağacı eğitilirken tercih edilen algoritma CART (Classification and Regression Tree) algoritmasıdır. Bu algoritma veri setini ikiye ayırarak ayrıştırmaya çalışır.

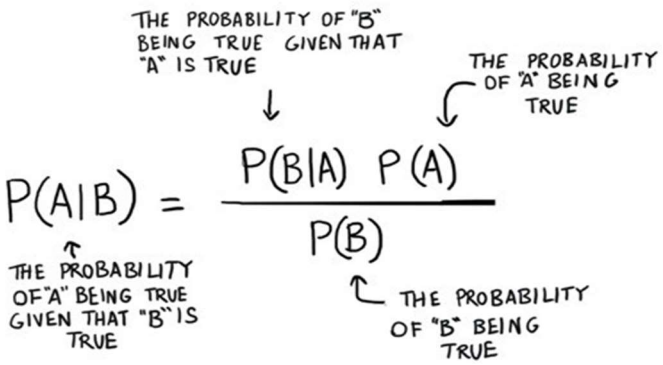
Şekil 18'de Karar Ağacı modelinin doğruluk oranı %99 olarak görülmektedir.

```
In [30]: accuracy_score(y_test, y_pred)
Out[30]: 0.9986633096170984
```

Şekil 18, Decision Tree Accuracy

Naive Bayes

Bayes teoremi, olasılık kuramı içinde incelenen önemli bir konudur. Bu teorem bir rassal değişken için olasılık dağılımı içinde koşullu olasılıklar ile marjinal olasılıklar arasındaki ilişkiyi gösterir. (https://tr.wikipedia.org/wiki/Bayes_teoremi, Erişim Tarihi: 04.06.2021)

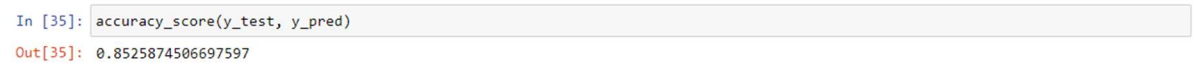


Şekil 19, Bayes Formülü

P(A) = A olayının gerçekleşme olasılığı
P (B) = B olayının gerçekleşme olasılığı
P(A|B) = B olayı gerçekleştiğinde A olayının gerçekleşme olasılığı
P(B|A) = A olayı gerçekleştiğinde B olayının gerçekleşme olasılığı

Algoritmanın çalışma şekli bir eleman için her durumun olasılığını hesaplamaya ve olasılık değeri en yüksek olana göre sınıflandırmaya dayanır. Az bir eğitim verisiyle çok başarılı işler çıkartabilir. Projede tercih edilme sebebi ağ trafiği normal veya kötü niyetli olarak sınıflandırmaya çalışılırken Bayes sınıflandırıcısını kullanmanın uygun olmasıdır.

Şekil 20'de Naive Bayes Modelinin doğruluk oranı %85 olarak görülmektedir.



Şekil 20, Naive Bayes Accuracy

Random Forest Algoritması

Rastgele orman algoritması, gözetimli bir makine öğrenmesi algoritmasıdır. Rastgele orman algoritması, hiperparatmetre ayarı yapılmadan iyi sonuçlar vermesi ve hem sınıflandırma hem regresyon problemlerine uygulanabilmesiyle en çok tercih edilen algoritmalardan biridir. Rastgele ormanın en büyük avantajı hem sınıflandırma hem regresyon problemlerinde kullanılabilmesidir.

Rastgele orman algoritmasının anlaşılabilmesi için Karar ağaçlarının anlaşılması gerekmektedir. Karar ağaçlarının en büyük problemlerinden biri aşırı öğrenmedir (overfitting). Rastgele orman algoritması bu problemin iyileştirilmesi için kullanılan bir algoritmadır. Problemin çözülebilmesi için hem veri setinden hem öznitelik setinden rastgele olmak üzere farklı setler seçilir bu setleri eğitilir. Bu yöntemle yüzlerce karar ağacı oluşturulur ve her bir karar ağacı kendi içinde tahminde bulunur. Son olarak eğer problemimiz bir regresyon ise karar ağaçlarından elde edilen tahminler toplanarak ortalaması alınır, eğer problemimiz bir sınıflandırma ise tahminler arasında en çok oy alan seçilir.

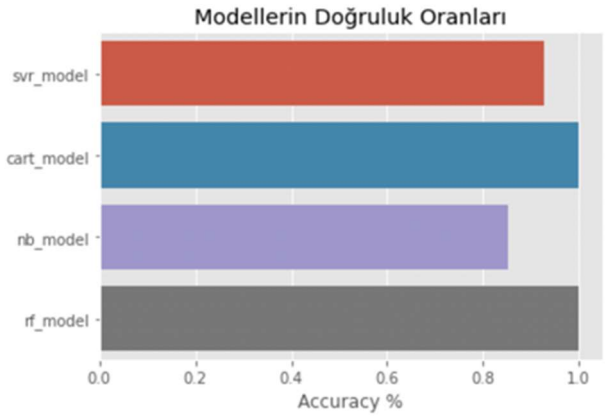
BÖLÜM 4
SONUÇ

Her geçen gün internetin yaygınlaşması ve buna bağlı olarak ağa bağlanan cihazların hızlı bir şekilde artması, bazı avantajlarının yanında birçok sorunu da beraberinde getirmektedir. Bu sorunlardan en önemlisi siber tehditlerdir. Kişilere, kurumlara ve devletlere karşı siber tehditler, maddi, itibar ve zaman gibi kayıplar verebilmektedir. Saldırı tespit ve saldırı önleme sistemleri, bu kayıpları ortadan kaldırmak veya en aza indirilebilmek için kullanılmaktadır. Saldırı tespit sistemleri imza tabanlı veya anomali tabanlı olarak tasarlanmakta ve günümüzde anomali tabanlı sistemler makine öğrenmesi yöntemleri kullanılarak geliştirilmektedir. Bu çalışmanın amacı, bir bilgisayar ağına saldırı olup olmadığını yüksek başarı oranı ile tespit etmektir. Bu sistemi geliştirmek için makine öğrenmesi yöntemlerinden Destek Vektör Makinesi (SVM- Support Vector Machine), Karar Ağacı (Decision Tree), Naive Bayes ve Random Forest kullanılmıştır. Sistemin geçerliliğini sınamak üzere CIC-IDS2017 veri seti kullanılmıştır.

Projemizde makine öğrenmesi modellerinin kullanılabilmesi için öncelikle kullandığımız veri setinin bazı ön işlemlerden geçmesi gerekmiştir. Öncelikle eksik verilerin tespiti ve veri setinden çıkarılması işlemi uygulanmıştır ardından verilerin eğitim ve test olarak ayrılması işlemi gerçekleştirilmiştir. Veri seti içindeki değerler herhangi bir makine öğrenmesi algoritmasına girmeden önce değerlerin belli bir aralık içerisine getirilmesi gerekir. Bu işlem Özellik Ölçekleme işlemidir. Veriler eğitim ve test olarak ayrıldıktan sonra bu işlem uygulanmıştır.- Veri setindeki değerler belli bir aralığa çekildikten sonra Öznitelik seçimi işlemi yapılmıştır. Öznitelik seçimi yöntemlerinden Ki-Kare yöntemi uygulanmıştır. Bu işlem gereksiz özniteliklerin veri setinden çıkarılması için kullanılmaktadır.

Projemizde yapılan tüm ön işlemlerin ardından çeşitli makine öğrenmesi modelleri uygulanmıştır. Bu modellerin her biri farklı performans özelliklerine sahiptir. Modellerin performans değerlendirmesi Doğruluk (Accuracy), Kesinlik (Precision), Recall, F1-Score gibi performans ölçütleri ile yapılabilir. Modeller birbirleri ile kıyaslanırken de yine bu ölçütler kullanılır. Bu ölçütler görselleştirilerek veri seti üzerinde hangi modelin nasıl performans gösterdiği daha kolay anlaşılabilir. Python'da scikit-learn ile bu görselleştirmeler yapılabilmektedir.

Aşağıdaki örnekte, kullandığımız dört farklı algoritma için doğruluk oranları karşılaştırılmıştır: Destek Vektör Makinesi (SVM- Support Vector Machine) Karar Ağacı (Decision Tree) Naive Bayes Random Forest.



Şekil 21, Modellerin Doğruluk Oranları

Şekil 21'deki grafikten de anlaşılacağı üzere rf_model olarak adlandırılan Random Forest modeli ile cart_model olarak adlandırılan Decision Tree modeli birbirine çok yakın ve en doğru sonuçları üretmişlerdir. Bu ölçümler 0.2 test verisi ile 0.8 eğitim verisi oranıyla hesaplanmıştır.

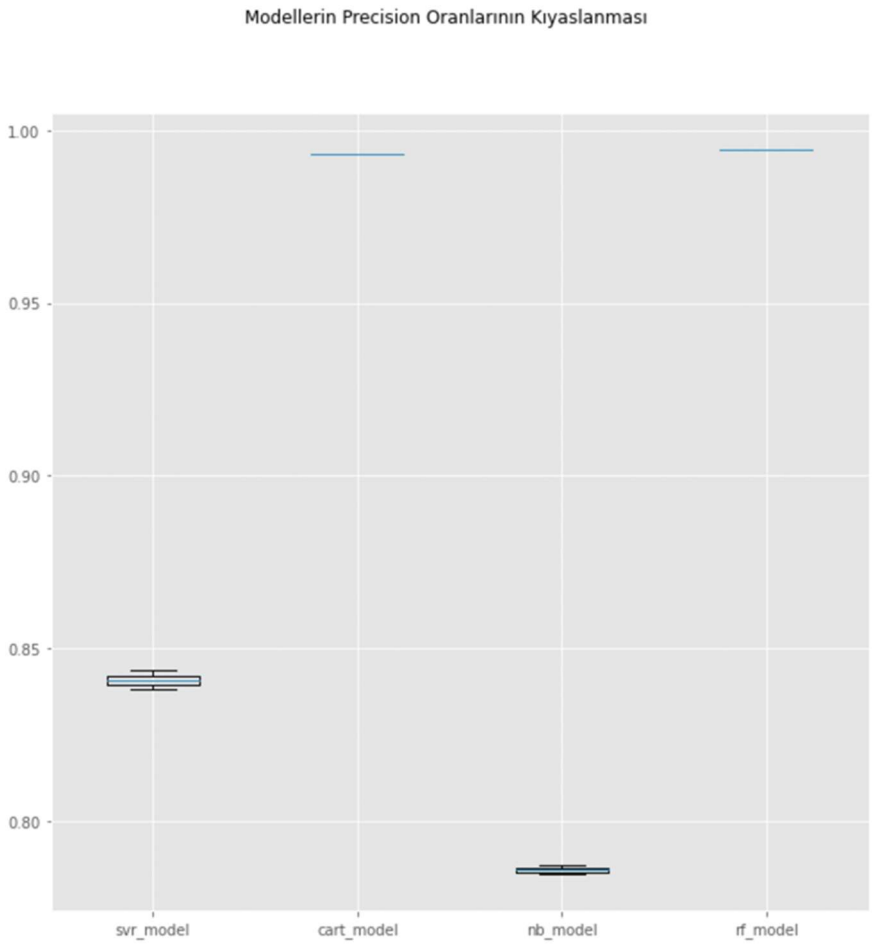
Decision Tree Accuracy: 0.9975016001782298
Random Forest Accuracy: 0.9973849020973827
SVM Accuracy: 0.9263776562074538
Naive Bayes Accuracy: 0.8652897471895212

Bu sonuçlara göre doğruluk oranı en yüksek olan model Decision Tree iken, en düşük olan model Naive Bayes modelidir.

Modellerin birbirlerine göre performans değerlendirmesini yapmak adına kullanılabilecek diğer bir yöntem precision oranlarının incelenmesidir. Precision (Kesinlik), yapılan tahminlerin yüzde kaçının doğru olduğunu ifade eder. Eğer precision (kesinlik) oranı

düşükse bu false-positive oranının yüksek olduğu manasına gelir. False-positive, saldırı olduğu halde saldırı değil olarak tespit edilen verileri temsil etmektedir ve false-positive oranı ne kadar fazla ise sistemin kullanılabilirliği o kadar düşük olur.

Precision = TP/(TP + FP)



Şekil 22, Modellerin Kesinlik Oranları

Şekil 22'deki grafikten de anlaşılacağı üzere rf_model olarak adlandırılan Random Forest modeli ile cart_model olarak adlandırılan Decision Tree modeli birbirine çok yakın ve en doğru sonuçları üretmişlerdir. Bu ölçümler 0.2 test verisi ile 0.8 eğitim verisi oranıyla hesaplanmıştır.

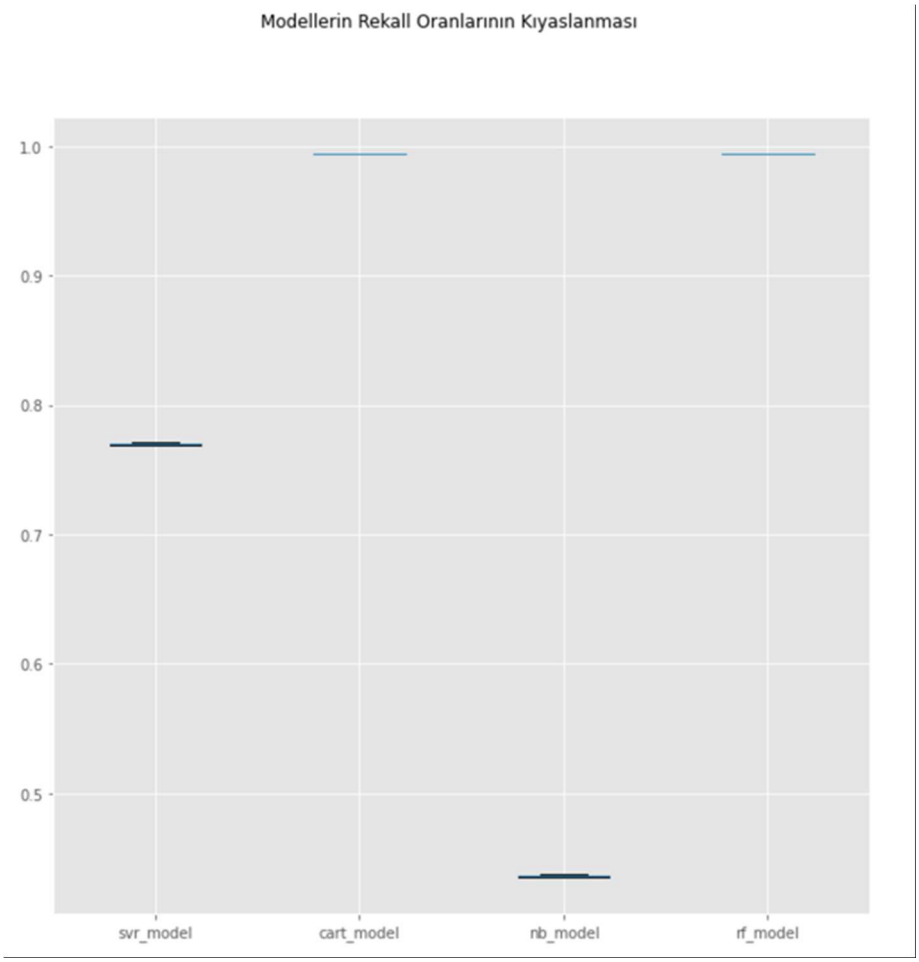
Decision Tree Precision score: 0.9929280675742463
Random Forest Precision score: 0.9903149648548586
SVM Precision score: 0.8418924104814666
Naive Bayes Precision score: 0.7834176886639413

Bu sonuçlara göre precision (kesinlik) oranı en yüksek olan model Decision Tree iken, en düşük olan model Naive Bayes modelidir.

Modellerin birbirlerine göre performans değerlendirmesini yapmak adına kullanılabilecek üçüncü yöntem ise recall oranlarının incelenmesidir. Recall, saldırıların yüzde

kaçının tespit edildiğini ifade eder. Düşük recall oranı sonuçlarda çok fazla false-negative olduğunu gösterir. False-negative saldırı olmadığı halde saldırı var denildiği durumdur.

Recall = TP/(TP+FN)



Şekil 23, Modellerin Recall Oranları

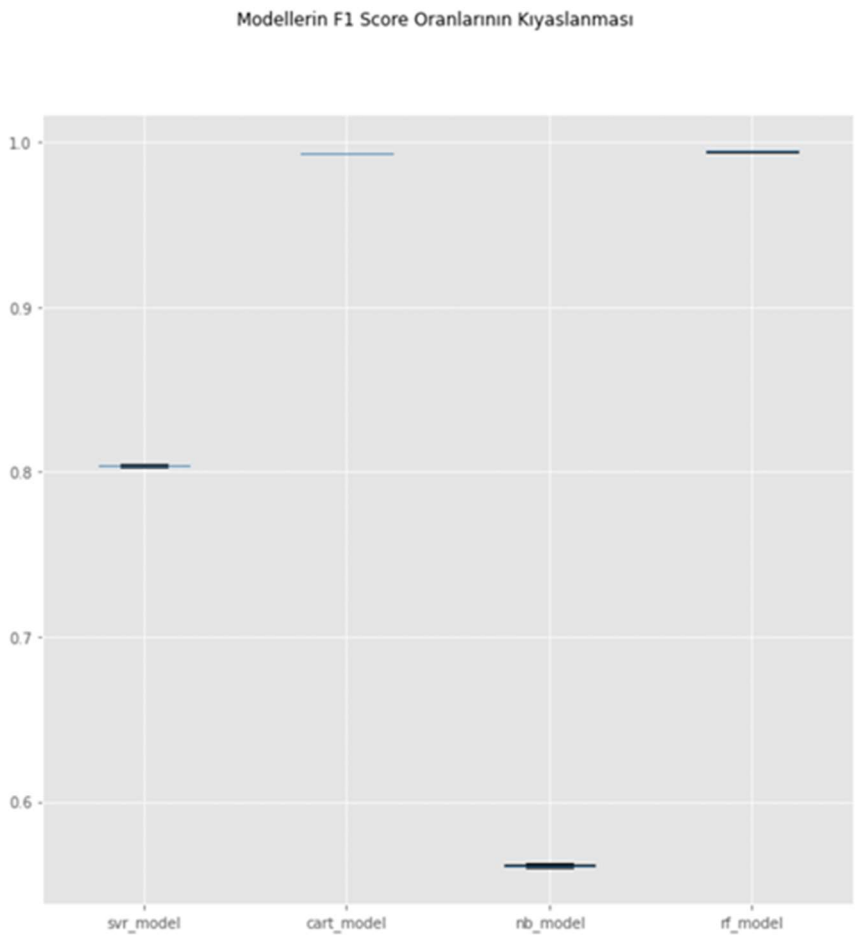
Şekil 23'teki grafikten de anlaşılacağı üzere rf_model olarak adlandırılan Random Forest modeli ile cart_model olarak adlandırılan Decision Tree modeli birbirine çok yakın ve en doğru sonuçları üretmişlerdir. Bu ölçümler 0.2 test verisi ile 0.8 eğitim verisi oranıyla hesaplanmıştır.

Decision Tree Recall score: 0.9943219141718184
Random Forest Recall score: 0.9964905650190319
SVM Recall score: 0.7699070449657606
Naive Bayes Recall score: 0.4345670347074121

Bu sonuçlara göre recall oranı en yüksek olan model Random Forest iken, en düşük olan model Naive Bayes modelidir.

Modellerin birbirlerine göre performans değerlendirmesini yapmak adına kullanılabilecek dördüncü yöntem ise F1 skor oranlarının incelenmesidir. F1 skoru, olumlu tahminlerin yüzde kaçının doğru olduğunu ifade eder.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)



Şekil 24, Modellerin F1 Skor Oranları

Şekil 24'teki grafikten de anlaşılacağı üzere rf_model olarak adlandırılan Random Forest modeli ile cart_model olarak adlandırılan Decision Tree modeli birbirine çok yakın ve en doğru sonuçları üretmişlerdir. Bu ölçümler 0.2 test verisi ile 0.8 eğitim verisi oranıyla hesaplanmıştır.

Decision Tree F1 Score: 0.9936245020547269
Random Forest F1 Score: 0.9933931671084677
SVM F1 Score: 0.8042922408768732
Naive Bayes F1 Score: 0.5590341083386872

Accuracy (doğruluk), Precision (kesinlik), Recall ve F1 skor değerleri göz önünde bulundurulduğunda Decision Tree (Karar Ağacı), Random Forest, SVM ve Naive Bayes modellerinden en yüksek doğruluk değerini ve en doğru tahminleri üreten model Decision Tree (Karar Ağacı) modeli olmuştur.

KAYNAKLAR

Prof. Dr. Vasif NABİYEV, Teoriden Uygulamalara Algoritmalar, 5. Basım, 2016, s. 94

Md Nasimuzzaman Chowdhury ve Ken Ferens, Mike Ferens, Network Intrusion Detection Using Machine Learning, 2016

Taner TUNCER, Yetkin TATAR, Karar Ağaçı Kullanarak Saldırı Tespit Sistemlerinden Performans Değerlendirilmesi, 2019

Yong Zhang, Xu Chen, Leı Jın, Xıaojuan Wang, Da Guo, Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data, 2019

XuKui Li, Wei Chen, Qianru Zhang, Lifa Wu, Building Auto-Encoder Intrusion Detection System based on random forest feature selection, 2020

https://github.com/kyralmozley/ids Erişim Tarihi:10.02.2021

https://www.geeksforgeeks.org/intrusion-detection-system-using-machine-learning-algorithms/ Erişim Tarihi: 01,05.2021

https://tr.wikipedia.org/wiki/Makine_öğrenimi - Erişim Tarihi: 01.05.2021

https://medium.com/cuelogic-technologies/evaluation-of-machine-learning-algorithms-for-intrusion-detection-system-6854645f9211 Erişim Tarihi: 01.05.2021

https://medium.com/deep-learning-turkiye/nedir-bu-destek-vektör-makineleri-makine-öğrenmesi-serisi-2-94e576e4223e Erişim Tarihi: 01.05.2021

http://www.firatipek.com/entry/15 Erişim Tarihi: 03.05.2021

https://numpy.org/ Erişim Tarihi: 03.05.2021

https://teknoloji.org/pandas-nedir-nasil-kullanilir-python-kutuphanesi/ Erişim Tarihi: 05.05.2021

https://pandas.pydata.org/ Erişim Tarihi: 05.05.2021

https://www.unb.ca/cic/datasets/ids-2017.html Erişim Tarihi: 05.05.2021

https://womaneng.com/imputation-yontemleri/s Erişim Tarihi: 05.05.2021

https://scikit-learn.org/stable/ Erişim Tarihi: 09.05.2021

https://veribilimcisi.com/2017/07/18/ozellik-olcekleme-ve-normallestirme-nedir-feature-scaling-and-normalization/ Erişim Tarihi: 09.05.2021

https://ng-dasci.medium.com/feature-scaling-nedir-1fbcd5cd125e Erişim Tarihi: 10.05.2021

https://tr.linkedin.com/pulse/özellik-seçme-yöntemleri-ml-iskender-yilmaz-msc Erişim Tarihi: 11.05.2021

https://yigitsener.medium.com/makine-öğrenmesinde-değişken-seçimi-feature-selection- yazı-serisi-filtreleme-yöntemleri-ve-415a894d5b93 Erişim Tarihi: 12.05.2021
https://yigitsener.medium.com/makine-öğrenmesinde-değişken-seçimi- feature-selection-yazı-serisi-sarmal-wrapper-yöntemler-ve-dd3b99c6c372 Erişim Tarihi: 18.05.2021

https://yigitsener.medium.com/makine-öğrenmesinde-değişken-seçimi- feature-selection-yazı-serisi-gömülü- embedded- c23293915b39 Erişim Tarihi: 18.05.2021

https://medium.com/data-science-tr/makine-öğrenmesi-dersleri-5- bagging-ve-random-forest-2f803cf21e07 Erişim Tarihi: 25.05.2021

https://www.geeksforgeeks.org/intrusion-detection-system-using-machine-learning-algorithms/ Erişim Tarihi: 25.05.2021

https://medium.com/cuelogic-technologies/evaluation-of-machine-learning-algorithms-for-intrusion-detection-system-6854645f9211 Erişim Tarihi: 25.05.2021

https://kodedu.com/2014/05/naive-bayes-siniflandirma-algoritmasi/ Erişim Tarihi: 21.05.2021

https://medium.com/@ekrem.hatipoglu/machine-learning-clustering-kümeleme-k-means-algorithm-part-13-be33aeef4fc8, Erişim Tarihi: 30.05.2021

https://medium.com/@ekrem.hatipoglu/machine-learning-classification-k-nn-k-en-yakın-komşu-part-9-6f18cd6185d Erişim Tarihi: 30.05.2021

https://owasp.org/www-project-top-ten/, Erişim Tarihi: 30.05.2021

https://medium.com/deep-learning-turkiye/karar-ağaçları-makine-öğrenmesi-serisi-3-a03f3ff00ba5, Erişim Tarihi: 01.06.2021

https://owasp.org/www-project-top-ten/, Erişim Tarihi: 02.06.2021

https://www.slideshare.net/oguzhantas/destek-vektr-makineleri-support-vector-machine Erişim Tarihi: 02.06.2021

https://medium.com/deep-learning-turkiye/karar-ağaçları-makine-öğrenmesi-serisi-3-a03f3ff00ba5, Erişim Tarihi: 04.06.2021

https://tr.wikipedia.org/wiki/Bayes_teoremi, Erişim Tarihi: 04.06.2021

https://www.slideshare.net/oguzhantas/destek-vektr-makineleri-support-vector-machine 04.06.2021

https://www.svm-tutorial.com/ 04.06.2021

STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Projemiz anomali tabanlı saldırı tespit sistemlerinde kullanılan makine öğrenimi yöntemlerinin uygulanmasıdır. Makine öğrenmesi yöntemleriyle anomali tespiti yapabilen lisanslı ve açık kaynaklı yazılımlar vardır. Bizim projemiz saldırı tespit sistemlerine eklenebilecek bir anomali tespit yazılımıdır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Geleneksel saldırı tespit sistemleri önceden bilinen saldırıların tespit edilmesinde oldukça kullanışlı olmasına rağmen 0. Gün saldırılarına karşı savunmasızdırlar. Geleneksel saldırı tespit sistemlerine alternatif bir çözüm olarak makine öğrenmesi yöntemleriyle geliştirilen "Anomali Tabanlı Saldırı Tespit Sistemleri" kullanılması hususunda projemiz çözüm sağlamaktadır.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Bilgisayar Ağları Dersi, Bilgisayar Ağ Programlama Dersi, Algoritmalar Dersi. Bu üç dersten edindiğimiz bilgisayar ağları ve algoritma becerilerini projemizde kullandık.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

ISO/IEC 27001: Bilgi Güvenliği Yönetim Sistemi için gereklilikleri ortaya koyan bir standarttır. ISO/IEC 27039: Saldırı Tespit ve Önleme Sistemleri (IDS / IPS) ile ilgilidir.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

Projemizde açık kaynak kodlu özgür ve ücretsiz teknolojileri tercih ettik.

b) Çevre sorunları:

Projemiz herhangi bir çevresel sorun oluşturmamaktadır.

c) Sürdürülebilirlik:

Projemiz minimum ve yeterli miktarda kaynak gereksinimleri üzerine inşa edilmiştir.

d) Üretilebilirlik:

Projemiz çapraz platformlarda çalışabilmektedir. Saldırı tespit sistemlerine entegre edilebilmektedir.

e) Etik:

Projemiz kullandığı teknolojiler dahilinde özgür ve açık kaynak kodlu bir sistemdir.

f) Sağlık:

Proje bileşenlerinin sağlık ile ilişkili bir tehlikesi bulunmamaktadır.

g) Güvenlik:

Projemizin ağ üzerindeki saldırıları tespit edebilmesi beklenmektedir. Var olan saldırıların tespit edilememesi güvenlik problemidir. Projenin fiziksel güvenlik problemi yoktur.

h) Sosyal ve politik sorunlar:

Güvenli bir sistem kurmak için bilgi güvenliği risklerini azaltmak ve etkisiz kılmak, iş süreçlerinin sürekliliğini sağlamak, olası maddi ve manevi kayıpları minimize etmek, kurumsal prestiji korumak, güvenlik sorumluluklarının ve bilincinin artmasını sağlamak gerekmektedir. Kişilere, kurumlara ve devletlere karşı siber tehditler, maddi zararların yanında itibar ve zaman gibi zararlara da sebep olabilmektedirler. Projemiz güvenliğin en iyi oranda sağlanabilmesini hedeflemektedir.