**IoT Cihazlarda Makine Öğrenimi Sınıflandırma Algoritmaları ile Ağ Saldırısı ve Anomali Tespiti**

İbrahim Mete Çiçek

Karabük Üniversitesi, Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği, Karabük, TÜRKİYE

**Özet**

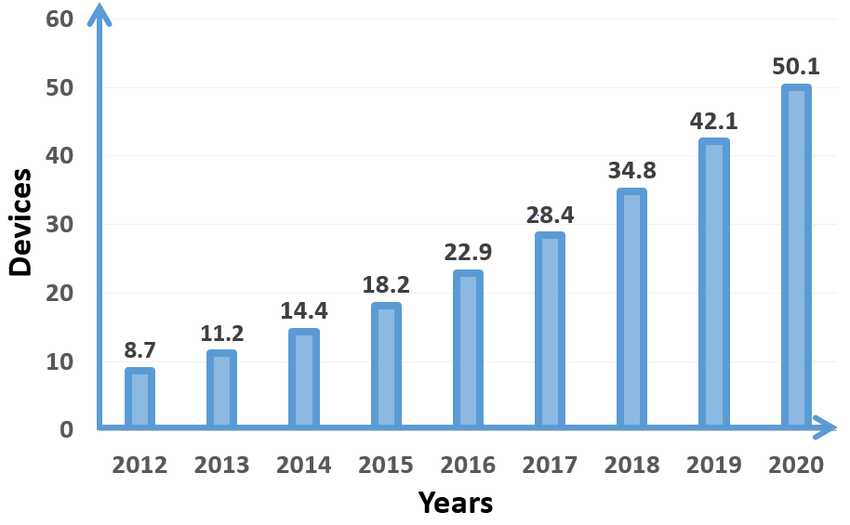
Günümüzde neredeyse bütün insanlar ve kurumlar internet üzerinden birbirleriyle veya uzaktaki cihazlarla iletişim kurmaktadır. Özellikle verinin öneminin yeni yeni anlaşılmaya başlandığı bu son zamanlara bağlı olarak internet üzerinden birçok cihazın ve büyük sistemlerin birbirine bağlı olduğu nesnelerin interneti denilen (IoT) iletişim ağındaki cihazlara yönelik yapılan saldırı vakalarının sayısı da buna paralel olarak artış göstermektedir. Yapılan bu saldırıların önüne geçmek için siber güvenlik alanında çok faktörlü kimlik doğrulaması ve imza tabanlı geleneksel yöntemler geliştirilip kullanılsa da sıfırıncı gün saldırılarına karşı başarısız olmaktadır. Fakat anomali tabanlı yaklaşım, sıfırıncı gün saldırılarının tespiti ve ağ saldırılarına karşı alternatif bir çözümdür. Bu çalışmada IoT cihazlara karşı yapılan saldırıları ve anomalileri doğru bir şekilde tahmin etmek için makine öğrenimi sınıflandırma algoritmaları kullanılmıştır. Örnek veri seti üzerinden önerilen yöntemin güçlü ve zayıf yönleri detaylı olarak incelenmiş ve performansları karşılaştırılarak ağ anomalilerinin ve saldırıların tespitinde elde edilen deneysel sonuçlar sunulmuştur.

**Anahtar Kelimeler:** IoT, Makine Öğrenimi, Ağ Anomali Tespiti, Siber Güvenlik

# Introduction

Dünya üzerinde her gün milyonlarca insan ve yüzbinlerce kurum internet üzerinden birbirleriyle iletişim kuruyor ve nesnelerin interneti dediğimiz (IoT) iletişim ağlarındaki cihazlar üzerinden birtakım işlemler gerçekleştiriyor. Öyle ki artık evlerde kullanılan lambalara varıncaya kadar her şey mobil cihazlardan veya internet servisleri üzerinden kontrol edilebilmektedir.

Bilindiği gibi son zamanlarda internet kullanan kişi sayısı çok hızlı artarken, günümüzde bu sayı 4 milyarın üzerine çıktı ve bu artış hızla devam etmektedir. Buna bağlı olarak kullanılan IoT cihazların sayısı da artmaktadır [1]. Şekil 1’de IoT cihaz sayısının yıllara göre gösterdiği artış verilmiştir.

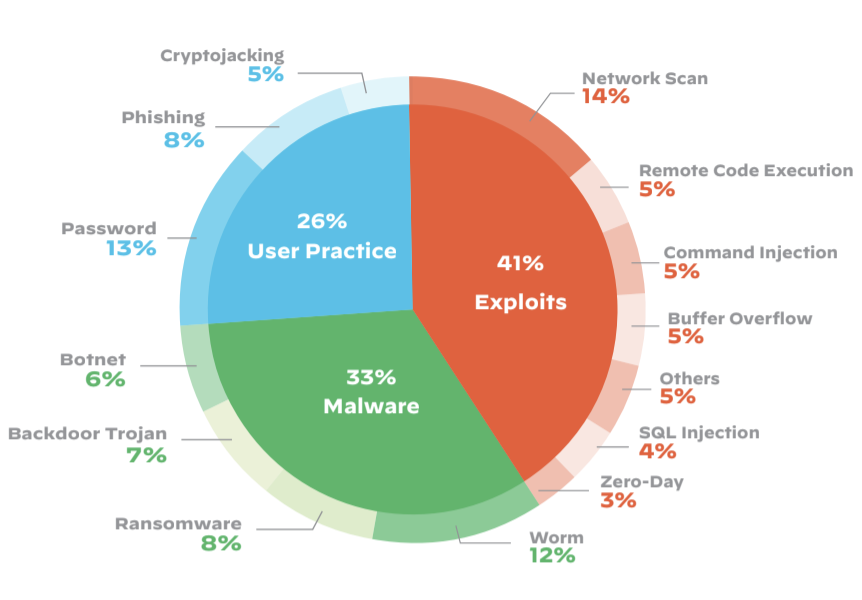


Şekil 1. IoT cihaz kullanıcı sayısının yıllara göre artışı (2012-2020) [2].

Bu gelişmeler insan hayatını kolaylaştırdıkça, yeni problemleri de beraberinde getirmektedir. Evlerde ve iş yerlerinde kullanılan akıllı cihazlar üzerinden özel ve iş hayatının gizliliğini ihlal eden durumlar söz konusu olabilmektedir.

İnternet altyapısı üzerinden izinsiz olarak başkalarına ait cihazların kontrol edilmesi, bilgilerine erişilmesi siber güvenliğin çalışma alanıdır. Akıllı cihazlarla birlikte siber güvenlik nesnelerin interneti alanında da önem kazanmıştır.

Şekil 2’de IoT cihazlara karşı yapılan siber saldırı yöntemlerinin dağılımı görülmektedir.



Şekil 2. IoT cihazlara karşı yapılan en yaygın siber saldırı yöntemlerinin oranları (2020) [3].

Bu tür siber saldırılara karşı, bilgi güvenliğini sağlamak için önce saldırıları tespit etmek gerekmektedir. Bunun için kullanılan 2 temel saldırı tespit sistemi (IDS) vardır; imzaya dayalı tanımlama (SIDS) ve anomaliye dayalı algılamadır (AIDS) [4].

İmza tabanlı yöntemler, saldırıları tespit etmek için oluşturdukları veri tabanını kullanır. Bu yöntem oldukça başarılıdır, ancak veri tabanlarının sürekli güncellenmesi ve yeni saldırı bilgilerinin işlenmesi gerekir. Ayrıca veri tabanları güncel olsa bile sıfırıncı gün dediğimiz önceden görülmemiş yeni saldırılara karşı savunmasızdır. Bu saldırılar veri tabanında olmadığı için maalesef algılayamazlar. Öbür taraftan bakacak olursak anomaliye dayalı yaklaşım ise direkt ağ akışını inceleyerek olağandışı ağ davranışlarını tespit etmeye odaklanır. Daha önce karşılaşmadığı saldırıları tespit etmede başarılı olan bu yöntem sonuç itibariyle sıfırıncı gün saldırılarına karşı da etkilidir denilebilir.

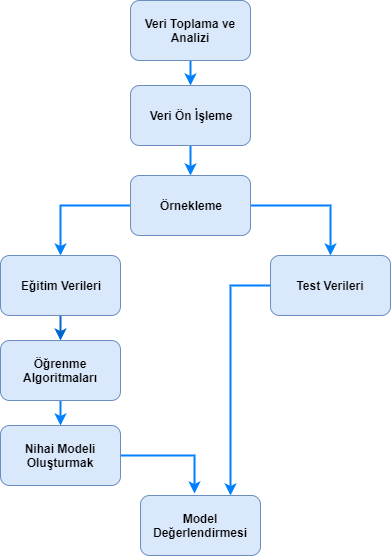
Günümüz internet kullanımının %50'sinden fazlası Güvenli Giriş Katmanı (SSL) ve Taşıma Katmanı Güvenliği (TLS) protokolleri kullanılarak şifrelenmektedir ve bu oran her geçen gün artmaktadır [5]. Şifrelenmiş internet akışının içeriğinin gözlemlenememesi nedeniyle, imza tabanlı yöntemler bu tür veriler üzerinde etkili bir şekilde çalışmaz. Bununla birlikte, anomaliye dayalı yaklaşım, verileri boyut, bağlantı süresi ve paket sayısı gibi genel özelliklerine göre analiz eder. Böylece mesaj içeriğini görmesine gerek olmadan şifreli protokollerin analizini de yapabilir. Tüm bu avantajlardan dolayı, ağ saldırılarını tespit etmek ve önlemek için anomali tabanlı tespit yöntemi yoğun bir şekilde kullanılmaktadır [6].

Bu çalışmada, ağ anomalilerini tespit etmek için çeşitli makine öğrenimi algoritmaları incelenecektir. Makine öğrenimi yöntemleriyle ağ anomalisini tespit ederek ağ saldırılarını hızlı ve etkili bir şekilde önlemek için bir yaklaşım geliştirilecektir. Çalışmada elde edilen sonuçları daha önce bu alanda yapılan çalışmalarla karşılaştırarak literatürdeki konumu ve katkıları sunulacaktır. Kısaca özetlemek gerekirse makine öğrenmesi yöntemleriyle ağ anomalisini hızlı ve etkin bir şekilde tespit eden bir sistem geliştirilerek sıfırıncı gün saldırılarına karşı literatüre katkı sağlanması amaçlanmaktadır.

# Materials and Methods

Genel olarak proje, birkaç bağımsız sürecin bir kombinasyonudur. Şekil 3 ’de bu sistemin genel çalışması gösterilmektedir. Bu sistemin ilk süreci veri seti toplama ve analizidir. Bu süreçte veri seti toplanmış ve veri türlerini bulmak için detaylı bir şekilde analiz edilmiştir. Ayrıca veri seti üzerinde veri ön işlemesi yapılmıştır. Veri ön işleme, verilerin temizlenmesi, verilerin görselleştirilmesi, özellik mühendisliği ve vektörleştirme adımlarından oluşur.

Bu adımlar verileri öznitelik vektörlerine dönüştürür. Bu özellik vektörleri daha sonra eğitim ve test seti olarak oranlara böler. Eğitim seti, öğrenme algoritmasında kullanılmış ve bir optimizasyon tekniği ile bir model geliştirilmiştir. Bu çalışmada farklı sınıflandırıcılar ve optimizasyon teknikleri kullanmıştır. Model, farklı değerlendirme ölçütleri kullanılarak test setine göre değerlendirildi.



Şekil 3. IoT'de saldırı ve anomali tespiti için genel süreç.

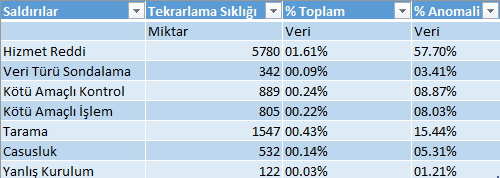
Veri Kümesi Toplama ve Açıklama

Makine öğrenimi yöntemleriyle ağ anomalisinin tespitinde, eğitim ve test adımları için büyük miktarda zararlı ve zararsız ağ trafiğine ihtiyaç vardır. Ancak, gizlilik sorunları nedeniyle gerçek ağ trafiğinin halka açık olarak kullanılması mümkün değildir. Bu ihtiyacı karşılamak için birçok veri seti üretilmiş ve üretilmeye devam etmektedir. Bu adımda bazı popüler veri kümeleri hakkında bilgi verilecek, ardından uygulama aşamasında hangilerinin kullanılacağına karar vermek için karşılaştırılacak ve değerlendirilecektir.

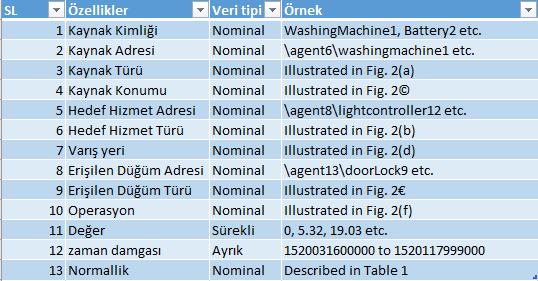
Açık kaynak veri seti, Pahl ve diğerleri tarafından sağlanan Kaggle'dan toplanmıştır. Sentetik veri üretmek için dağıtılmış akıllı alan düzenleme Sistemi kullanarak sanal bir IoT ortamı oluşturdular. Veri mimarileri, mesaj sıralı telemetri aktarım protokolünü kullanarak birbirleriyle iletişim kuran bir mikro hizmetler koleksiyonudur. [7]

Tablo 1, tüm veriler boyunca farklı saldırıların ve anomali dağılımının ayrıntılı bir resmini vermektedir. Veri setindeki 13 özellik açıklamaları ile birlikte Tablo 2'de verilmiştir.

Tablo 1. Değerlendirilen saldırıların frekans dağılımı.



Tablo 2. Özelliklerin açıklaması.



1. Hizmet Reddi: Bu saldırılar, tek bir kaynak veya alıcıda çok fazla istenmeyen trafiğin bulunmasından kaynaklanır. Saldırgan, hedefin band genişliğini aşmak ve sunulan servislerin diğer hizmetler için kullanılamaz hale getirmek için çok fazla belirsiz paket gönderir. Bu çalışmadaki veri setinde farklı birçok örnek DoS saldırısı içermektedir.

2. Veri Türü Manipülasyonu: Bu gibi durumlarda, kötü niyetli bir düğüm, amaçlanan veri türünden farklı veri türleri yazar. Bu çalışmadaki veri setinde farklı veri tipi manipülasyon örnekleri bulunmaktadır.

3. Kötü Amaçlı Denetim: Yazılım güvenlik açıklarında bazen saldırgan geçerli bir oturum anahtarı elde edebilir veya bir şekilde ağ trafiğini yakalayabilir. Bu şekilde kötü niyetli kişi tüm sistemi kontrol edebilir. Bu çalışmadaki veri seti birbirinden farklı kötü amaçlı kontrol örnekleri içermektedir.

4. Kötü Amaçlı İşlem: Kötü amaçlı işlemlere genellikle kötü amaçlı yazılımlar neden olur. Kötü amaçlı yazılım, orijinal işlemin dikkatini dağıtan oltalama etkinliği anlamına gelir. Cihazın performansı bu kötü niyetli işlemden olumsuz etkilenebilir. Bu çalışmadaki veri setinde farklı kötü amaçlı işlem örnekleri bulunmaktadır.

5. Tarama: Bazen veriler donanım aracılığıyla sistem taranarak alınır ve bu süreçte veriler bozulabilir. Bu çalışmadaki veri setinde birçok Tarama örneği içermektedir.

6. Casusluk: Saldırgan, casusluk yaparak sistemin güvenlik açıklarından yararlanır ve sisteme girmek için bir arka kapı kanalı kullanır ve önemli bilgileri keşfeder. Bazı durumlarda, verileri manipüle ederek tüm sistemde büyük engellere neden olurlar. Bu çalışmadaki veri setinde farklı Casusluk örnekleri içermektedir.

7. Yanlış Kurulum: Veriler, yanlış sistem kurulumu sonucunda da bozulabilir. Bu çalışmadaki veri setinde, yanlış kurulum örnekleri içermektedir.

8. Normal Veri: Tamamen doğru ve eksiksiz verilere normal veri denir. Bu çalışmadaki veri setinde, normal veri sınıfı bulunmaktadır.

Veri Ön İşlemesi

Herhangi bir makine öğrenimi araştırması, keşif amaçlı veri analizi ve gözlemi gerektirir. Bu araştırmadaki ilk görev, veri kümesini herhangi bir sınıflandırıcı için beslenebilir hale getirmektir. Bu nedenle, ilk adım eksik verileri ele almaktadır. Veri kümesinde, "erişilen düğüm türü" sütunu ve "değer" sütunu, veri aktarımında ortaya çıkan anomali nedeniyle eksik değerler içeriyor. Bu iki özellikten “erişilen düğüm tipi” özelliği kategorik değerlere sahipken, “değer” özelliği sürekli değerlere sahiptir. "erişilen düğüm türü" özelliği, sayı değil olarak gösterilen "NaN" değerini içeren satırlara sahiptir ve bu satırın karşılık gelen sınıf veya etiketinin anomali olduğu bulunmuştur. "erişilen düğüm türü" özelliği kategorik olduğundan, "erişilen düğüm türü"ndeki "NaN" değeri "kötü amaçlı" değeriyle değiştirilir. Benzer şekilde, “değer” sütunu da sürekli olmayan bazı beklenmedik veriler içerir. Bu beklenmedik değerler, sınıflandırıcıların daha iyi doğruluk elde etmesine yardımcı olan anlamlı sürekli değerlere dönüştürülür.

Özellik seçimi için burada Pahl ve diğerleri gibi hiçbir makine öğrenimi yaklaşımı alınmamıştır. Çünkü bunun veri analizi üzerinde önemli bir etkisi olmayacaktır. Ayrıca bu zaman damgası sütunu, veri kümesinin öngörücü değişken normalliği ile minimal bir korelasyona sahip olduğu için veri kümesinden kaldırılmıştır.

Özellik mühendisliği adımlarında, öncelikle veri setindeki özellik tipinin belirlenmesi gerekir. Veri kümesi kategorik ve sayısal verileri içerir. Kategorik veriler ayrıca sıralı ve nominal değerler olarak sınıflandırılabilirken, sayısal veri kümesi ayrık ve sürekli değerler olarak sınıflandırılabilir. Tablo 2, sütun türlerini göstermektedir. Dolayısıyla tablo 2'den “değer” sütunu ve “zaman damgası” sütunu dışındaki tüm sütunların kategorik nominal değişken olduğu iddia edilebilir. Ayrıca, "değer" sütunu ve "zaman damgası" sütunu sürekli sayısal değişkendir. "zaman damgası" sütunu, veri kümesinden kaldırıldığı için burada dikkate alınmamıştır.

Bir sonraki önemli görev, nominal kategorik verileri vektörlere dönüştürmektir. Kategorik veriler birçok şekilde vektörlere dönüştürülebilir. Bunlar arasında etiket ve sıcak kodlama yaygındır. Bu araştırmada, verileri bir öznitelik vektörüne dönüştürmek için etiket kodlama tekniği kullanılmıştır. Bu veri kümesindeki özelliklerin çoğu, nominal kategorik değer ve birçok benzersiz değer içerir. Bu özniteliklere bir sıcak kodlama uygulansaydı, özniteliklerin sayısı önemli bir sayıyla artacak ve elde edilen veri kümesinin birçok boyutu olacaktı. Diğer durumda, etiket kodlamasına göre, özelliklerin sayısı aynıydı. Böylece veri kümesinin boyutu artırılmadı. Bunların yanı sıra, bir sıcak kodlanmış özellik, makine öğrenimi algoritmasına sığması daha zor olan ve çok fazla işlem süresi alan seyrek özelliklere sahip olacaktır. Bu nedenle, veri kümesine etiket kodlaması uygulanır.

Makine Öğrenmesi

Makine öğrenimi, programlanmış bilgisayarların kendilerine verilen verilerden öğrenmesini sağlayan bir bilim ve sanattır. Makine öğrenmesi sürecinde bilgisayarlar kendilerine verilen veriler yani eğitim seti üzerinden eğitilebilmekte ve farklı bir veri yani test seti üzerinde performanslarını gösterebilmektedirler. Bu şekilde insan müdahalesi en aza indirilerek sorun çözülür. Klasik yöntemlerin verimsiz olduğu birçok yerde makine öğrenimi yoğun bir şekilde kullanılmaktadır. Kullanım alanları şu şekilde sıralanabilir:

• Çok büyük ve karmaşık verileri yorumlayabilir.

• Geleneksel yöntemlerin çözüm bulamadığı karmaşık sorunları çözebilir.

• Makine Öğrenimi yöntemleri, mevcut çözümlerin çok fazla güncelleme gerektirdiği durumlara dış müdahale olmaksızın çözüm bulabilir.

• Değişken ortamlarda çalışabilir. Makine öğrenimi yöntemleri, veriler incelenerek yeni bir duruma uygulanabilir.

Makine öğrenme algoritmaları, eğitim verilerinin etiketlenip etiketlenmemesine ve aldıkları eğitim süpervizyonuna göre 4 gruba ayrılır. Bunlar denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve pekiştirmeli öğrenmedir.[8]

Denetimli öğrenme: Bu yöntemde, eğitim verileri doğru şekilde sınıflandırılmış ve etiketlenmiştir. Örneğin, veri kümesindeki tüm akışlar, doğaları hakkında bilgiler içerir. Bir sonraki aşamada, tahmin aşamasında, bu etiketler algoritmanın bulduğu sonuçlarla karşılaştırılır ve algoritmanın başarısı hesaplanır. Yöntemin performansı yüksektir. Ancak denetimli öğrenme, etiketleme için harici hizmet kullandığından maliyetlidir. Bu tür algoritmalara örnek olarak Karar Ağaçları, K-En Yakın Komşular ve Rastgele Ormanlar verilebilir.

Denetimsiz öğrenme: Bu yöntemde bir etiketleme süreci yoktur. Algoritma, verileri çeşitli özelliklere göre gruplara ayırır ve birbirleriyle ilişkilerini gözlemler. Anomali tespiti, ilişki öğrenme ve boyut küçültme gibi alanlarda yaygın olarak kullanılmaktadır. Bu yöntemin maliyeti düşüktür çünkü etiketleme gibi dış kaynak uzmanlığı gerektirmez.

Yarı denetimli öğrenme: yöntem, denetimli öğrenme denetimsiz öğrenme yöntemiyle birleştirildiğinde ortaya çıkan karma bir yöntemdir. Genellikle, verilerin çok küçük bir kısmı etiketlenir ve geri kalanı etiketlenmez. Bu yöntem, denetimli öğrenmenin yüksek performansını denetimsiz öğrenmenin düşük maliyeti ile birleştirir.

Pekiştirmeli Öğrenme: Bu yaklaşım, temel ilke olarak diğer üçünden çok farklıdır. Bu sistemde, algoritma eğitim sırasında yapılan yanlış seçimler için ceza alır ve doğru seçimler için ödül alır. Böylece algoritma kendi kurallarını oluşturur.

Bu projede, manuel olarak etiketlenmiş iyi bir veri setine sahip olmanın avantajlarından yararlanmak için denetimli öğrenme yöntemleri kullanılacaktır. Bu sayede maliyet dezavantajı elde edilmeden yüksek performans avantajı elde edilmesi hedeflenmektedir.

Teorik Düşünceler

Veri analizi kısmı için çeşitli makine öğrenme algoritmaları kullanılmıştır. Aşağıda açıklamalarıyla birlikte algoritma listeleri bulunmaktadır.

eXtreme Gradyan Artırma (XGBoost)

XGBoost, analistlerin tahmin edilecek bir sonucu belirttiği, denetimli ML kullanan, kitaplık olarak adlandırılan bir dizi açık kaynak işlevi ve adımıdır. XGBoost kitaplığı, bir sonucu tahmin etmek için birden çok karar ağacı kullanır.

Makine öğrenimi sistemi toplu öğrenme kullanılarak eğitilir ve model tabanlı bir yaklaşımla genelleştirilir. Tahmin edici ve sonuç değişkenleri arasındaki ilişkiyi belirten bir model oluşturmak için mevcut tüm verileri kullanır ve bunlar daha sonra test verilerine genelleştirilir.

XGBoost, eXtreme Gradient Boosting anlamına gelir. “Aşırı” kelimesi, hesaplama kaynaklarının sınırını zorlama hedefini yansıtır. Gradyan artırma ise, doğru ve güvenilir bir tahmin edici oluşturmak amacıyla bir dizi zayıf tahmin modelini optimize eden, regresyon ve sınıflandırma problemleri için bir makine öğrenimi tekniğidir. [9]

CatBoost

Ayrıca, Karar Ağacı tabanlı bir algoritma olarak CatBoost, kategorik, heterojen veriler içeren makine öğrenimi görevleri için çok uygundur. Birden fazla disiplinde yapılan son çalışmalar, CatBoost'un sınıflandırma ve regresyon görevlerindeki etkinliğini ve eksikliklerini göstermektedir. CatBoost ile ilgili literatürde ortaya koyduğumuz bir diğer önemli konu, hiper parametrelere duyarlılığı ve hiper parametre ayarının önemidir. Yaptığımız katkılardan biri, CatBoost ile ilgili çalışmaları tek bir çalışmada ele almak için disiplinler arası bir yaklaşım benimsemektir. Bu, araştırmacılara, problemlerin çözümünde CatBoost'un doğru uygulamasını netleştirmeye yardımcı olacak derinlemesine bir anlayış sağlar. Bildiğimiz kadarıyla bu, CatBoost ile ilgili tüm çalışmaları tek bir yayında inceleyen ilk ankettir. [10]

Gradient Boosted

Karar Ağaçları (GBDT'ler), Büyük Veride sınıflandırma ve regresyon görevleri için güçlü bir araçtır. Araştırmacılar, onları etkin bir şekilde kullanmak ve başarılı katkılarda bulunmak için GBDT'lerin mevcut uygulamalarının güçlü ve zayıf yönlerine aşina olmalıdır. CatBoost, GBDT makine öğrenimi topluluğu teknikleri ailesinin bir üyesidir. 2018 yılının sonlarında piyasaya sürülmesinden bu yana araştırmacılar, Büyük Veri içeren makine öğrenimi çalışmaları için CatBoost'u başarıyla kullandılar. Bu fırsatı, Büyük Veri ile ilgili olduğu için CatBoost hakkındaki son araştırmaları gözden geçirmek ve CatBoost'u olumlu bir ışık tutan çalışmaların yanı sıra CatBoost'un diğer teknikleri gölgede bırakmadığı çalışmalardan en iyi uygulamaları öğrenmek için kullanıyoruz, çünkü her ikisinden de dersler çıkarabiliriz. [10]

AdaBoost

Bir boosting yöntemi olan AdaBoost (Adaptive Boosting), sınıflandırma performansını iyileştirmek için geliştirilmiş bir makine öğrenme algoritmasıdır. Boosting algoritmalarının temel çalışma prensibi şu şekilde açıklanabilir: Veriler önce kaba taslak kurallarla gruplara ayrılır. Algoritma her çalıştırıldığında bu kaba taslak kurallara yeni kurallar eklenir. Bu şekilde "temel kurallar" olarak adlandırılan birçok zayıf ve düşük performans kuralı elde edilir [11].

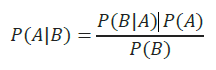
K En Yakın Komşu (KNN)

Örnek tabanlı bir yöntem olan K En Yakın Komşu, basit ve hızlı yapısı ile en çok kullanılan makine öğrenmesi algoritmalarından biridir. Bu algoritma, bir veri kümesindeki örneklerin başka bir veri kümesindeki benzer özelliklere sahip örneklere yakın olacağı varsayımına dayanır. [12]

Naive Bayes (NB)

Bayes teoremine bağımsızlık koşulunun eklenmesiyle basitleştirilmiş bir makine öğrenme algoritmasıdır [13].

Bayes teoremi aşağıdaki denklemle ifade edilir:



Lojistik Regresyon (LR)

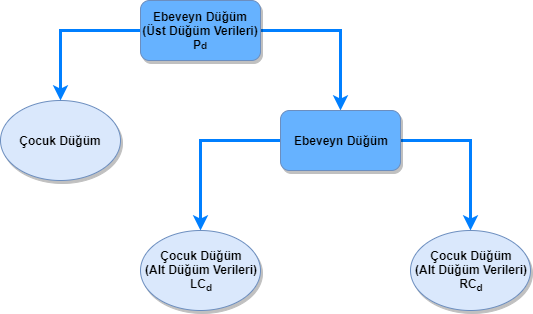
Lojistik Regresyon (LR), veri setinin kalitesine bağlı olan ayırt edici bir modeldir. [14]



Karar Ağacı (DT)

Karar Ağacı, her bir düğümün faydalarına, maliyetlerine ve olasılıklarına dayalı olarak olası eylemleri birbirleriyle karşılaştırmasına olanak tanır. Genel olarak, bir dizi ilgili seçeneğin olası sonuçlarının bir haritasıdır. Bir DT genellikle tek bir düğümle başlar ve ardından olası sonuçlara dallanır. Bu sonuçların her biri, diğer örneklere ayrılan ek düğümlere yol açar. Böylece oradan ağaca benzer bir şekil oldu; başka bir deyişle, akış şeması benzeri bir yapı. Bir ana düğümün, sol çocuk ve sağ çocuk olmak üzere iki alt düğüme ayrıldığı bir ikili ağaç düşünüldüğünde. Ana düğüm, sol çocuk ve sağ çocuk sırasıyla Pd, LCd, RCd verilerini içerir. Verilen, özellikler x, safsızlık ölçüsü I, ana düğüm Pn'deki örnek sayısı, sol çocuk LCn'deki örnek sayısı ve sağ çocuk RCn'deki örnek sayısı; DT’nin hedefi, aşağıdaki Denklemdeki Bilgi Kazancını maksimize etmektir. [15]





Şekil 4. Karar ağacının yapısı ve çalışma şekli.

Rastgele Orman (RF)

Adından da anlaşılacağı gibi, rastgele orman algoritması, birçok karar ağacı içeren ormanı oluşturur. Denetimli bir sınıflandırma algoritmasıdır. Yüksek yürütme hızı nedeniyle çekici bir sınıflandırıcıdır. Pek çok karar ağacı rastgele bir orman oluşturmak için bir araya gelir ve her bir bileşen ağacın tahminlerinin ortalamasını alarak öngörüde bulunur. Genellikle tek bir karar ağacından çok daha iyi tahmin doğruluğuna sahiptir. Genel olarak, ormandaki ağaç sayısı ne kadar fazlaysa orman o kadar sağlam görünür. [16]

Yapay Sinir Ağı (ANN)

MLP (Çok Katmanlı Algılayıcı), yapay sinir ağlarının bir türüdür. Yapay sinir ağları (YSA), insan beyninin çalışma şeklinden ilham alan bir makine öğrenme yöntemidir. Bu yöntemin amacı, öğrenme, karar verme ve yeni bilgi türetme gibi insan beyninin özelliklerini taklit etmektir. İnsan beyni nöron adı verilen birbirine bağlı hücrelerden oluşurken, yapay sinir ağları birbirine bağlı hiyerarşik yapay hücrelerden oluşur. [17] Bu projede yapay sinir ağı yaklaşımı olan MLP kullanılmıştır.



Değerlendirme Kriterleri

Geliştirilen sistemin performansını değerlendirmek için aşağıdaki metrikler hesaplanmıştır. Bu ölçütleri kullanarak, bu iş için hangi tekniğin en uygun olduğuna karar verilebilir.

Karışıklık Matrisi

Karışıklık matrisi, bir tekniğin performansını görselleştirmek için kullanılır. Gerçek değerlerin bilindiği bir dizi test verisi üzerinde bir sınıflandırma modelinin performansını tanımlamak için sıklıkla kullanılan bir tablodur. Sınıflar arasındaki karışıklığın kolayca tanımlanmasını sağlar. Çoğu zaman, neredeyse tüm performans ölçümleri ondan hesaplanır. Karışıklık matrisi, bir sınıflandırma problemine ilişkin tahmin sonuçlarının bir özetidir. Karışıklık matrisinden birden fazla sınıf için True Positive (TP), False Positive (FP), False Negative (FN) ve True Negative (TN) tanımları verilebilir. Ci, sekiz sınıftan herhangi bir sınıf olsun. [18]

Doğruluk

Bir modelin doğruluğu, modelin performansının yalnızca bir alt kümesidir. Doğruluk, sınıflandırma modellerini değerlendirmek için kullanılan metriklerden biridir. Denklem, tek sınıf doğruluk ölçümünü tasvir etmektedir. [15]



Kesinlik

Kesinlik, pozitif tahmin değeri anlamına gelir. Modelin iddia ettiği pozitiflerin sayısına kıyasla iddia ettiği gerçek pozitiflerin sayısının bir ölçüsüdür. Tek bir sınıf için kesinlik değeri aşağıdaki denklemde verilmiştir: [16]



Geri Çağırma

Geri çağırma, gerçek pozitif oran olarak bilinir; bu, model iddialarındaki pozitiflerin sayısının, veriler boyunca var olan gerçek pozitiflerin sayısına kıyasla anlamına gelir. Tek bir sınıf için geri çağırma değeri aşağıdaki denklemde verilmiştir: [17]



F1 Puanı

F1 puanı, bir modelin performansını da ölçebilir. Bir modelin kesinliğinin ve geri çağrılmasının ağırlıklı ortalamasıdır. Tek bir sınıf için F1 Puanı değeri Denklem'de verilmiştir. [18]



Alıcı Çalışma Karakteristik Eğrisi (ROC)

Bir sınıflandırıcının performansını tüm olası eşikler üzerinde özetleyen yaygın olarak kullanılan bir grafiktir. Belirli bir sınıfa gözlemleri atamak için eşiğin değeri değiştiğinden, Gerçek Pozitif Oranın Yanlış Pozitif Hıza karşı grafiğinin çizilmesiyle oluşturulur. Gerçek Pozitif Oran ve Yanlış Pozitif Oranın hesaplanması aşağıdaki denklemlerde verilmiştir: [19]





Eşik değeri, tahmin edilen her sınıf için olasılık değeridir. ROC eğrisi, ikili sınıflar kullanılarak çizilebilir. Ancak, bire karşı dinlenme yöntemi kullanılarak birden çok sınıf için genişletilebilir. Her sınıf için gerçek pozitif oran ve yanlış pozitif oran değerleri 0 ile 1 arasında değişir.

# Implementation

Araçlar ve Yöntemler Yazılım Platformu

Ücretsiz ve açık kaynak kodlu nesne yönelimli bir programlama dili olan Python, basit sözdizimi ve dinamik yapısıyla dikkat çekiyor. Python'da kod yazmak ve kodu analiz etmek çok kolaydır. Diğer bir avantajı ise kapsamlı dokümantasyon avantajına sahip olmasıdır. Tüm bu avantajlarının yanı sıra "makine öğrenimi" uygulamalarının yapılabileceği birçok kütüphane ile uyum içinde çalışmaktadır. Bu bağlamda, sağladığı birçok avantajdan dolayı bu çalışmada kullanılmak üzere Python 3.9 seçilmiştir. [20]

Sklearn (Scikit-learn), Python programlama dili ile kullanılabilen bir makine öğrenme kitaplığıdır. Sklearn, çok sayıda makine öğrenmesi algoritması ile kullanıcıya geniş bir seçenek yelpazesi sunar. Sklearn kapsamlı belgelere sahiptir ve bu iş için gereken tüm algoritmaları içerir.

Pandas, Python üzerinde çalışan güçlü bir veri analiz kütüphanesidir. Pandas, büyük bir veri seti ile çalışırken filtreleme, toplu sütun ve satır silme, ekleme, değiştirme gibi birçok işlemi kolaylıkla gerçekleştirmenize olanak sağlar. Tüm bu avantajlardan dolayı Pandas kütüphanesi kullanılmıştır.

Matplotlib, Python üzerinde çalışan ve verilerin görselleştirilmesine izin veren bir kütüphanedir. Bu kütüphane, çalışmada kullanılan grafikleri oluşturmak için kullanılır.

Bu çalışmada hesaplamalarda matematiksel ve mantıksal işlemleri hızlı ve kolay bir şekilde yapmanızı sağlayan bir Python kütüphanesi olan NumPy kullanılmıştır.

Donanım Platformu

Makine öğrenimi algoritmaları için bir değerlendirme kriteri yürütme süresidir. Ancak, yürütme süresi, kullanılan bilgisayarın performansına bağlı olarak değişebilir. Bu nedenle uygulamada kullanılan bilgisayarın teknik özellikleri paylaşılmıştır. Uygulama aşamasında kullanılan bilgisayarın teknik özellikleri şunlardır:

Merkezi işlem birimi: Intel (R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz

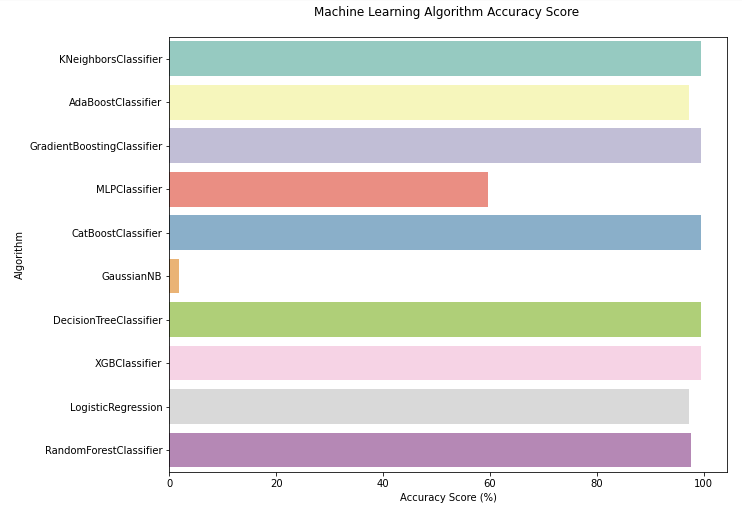
Rasgele erişim belleği: 8,00 GB

İşletim sistemi: Windows 10 Pro 64 bit

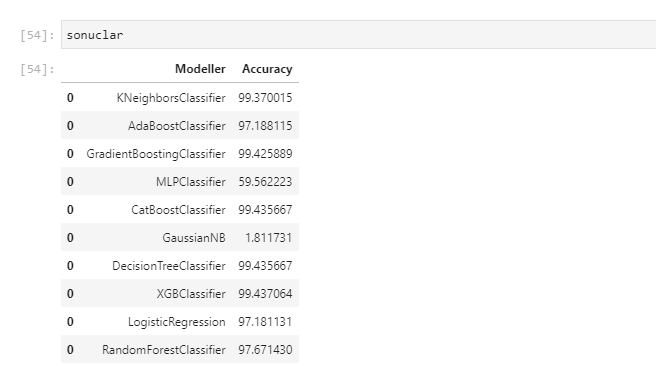
Grafik İşleme Ünitesi: Intel® HD Graphics 620

Sonuç Analizi

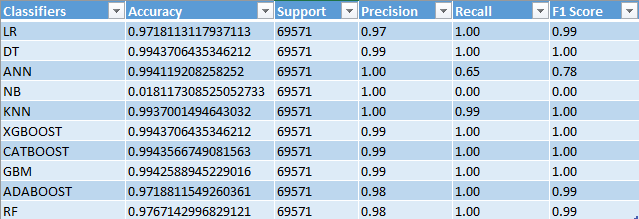
Veri Analizi alt bölümünde, veri kümesine birkaç makine öğrenimi tekniğinin uygulandığı açıklanmıştır. Bu tekniklerin her biri kullanılarak veri setinde beş katlı çapraz doğrulama yapıldı. Çapraz doğrulamadan, XGBClassifier’ın hem eğitimde hem de test doğruluğunda en iyi performansı gösterdiği sonucuna varılabilir. Test durumunda DT, GradientBoosting, KNeighborsClassifier ve CatBoostClassifier'a yaklaşık benzerlikle gerçekleştirilmiştir. Tablo 3, veri kümesi üzerinde eğitilen farklı teknikler için farklı değerlendirme ölçütlerini temsil etmektedir. Tablo 3'ten DT ve XGBOOST'un diğer tekniklere göre daha fazla doğruluk, kesinlik, geri çağırma ve F1 skor değerlerine sahip olduğu görülmektedir.



Şekil 5. Kullanılan sınıflandırıcıların doğruluk değerlerinin grafiksel gösterimi.

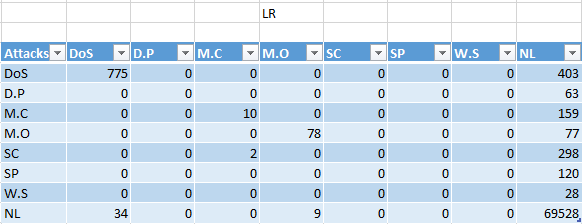


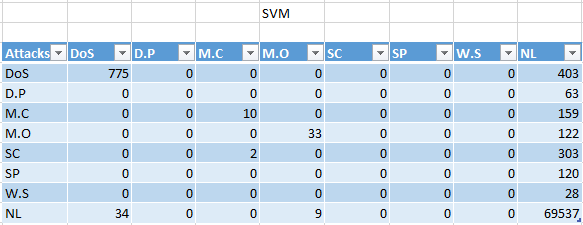
Şekil 6. Kullanılan sınıflandırıcıların doğruluk tablosu.

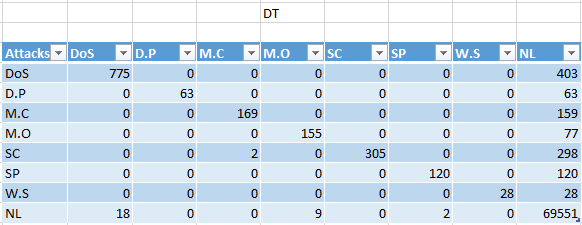


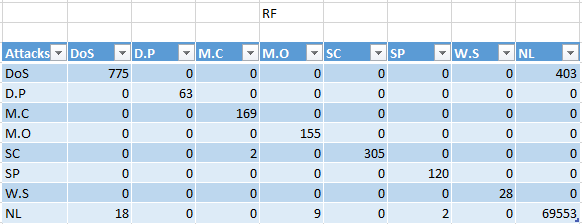
Tablo 3. Çalışmamızın değerlendirme ölçütleri.

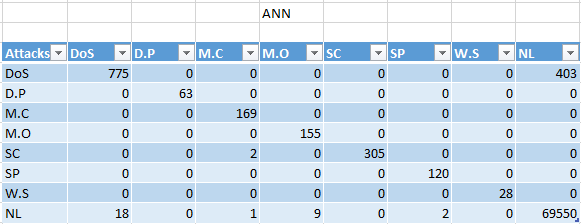
Şimdi her bir tekniğin karışıklık matrisleri göz önüne alındığında, en optimize edilmiş teknik bulunabilir. Şekil 7'teki karışıklık matrislerinden, XGBOOST'un bu iş için en iyi teknik olduğu sonucuna varılabilir. XGB, DoS ve Normality sınıfları dışında her sınıfı doğru bir şekilde sınıflandırdı.





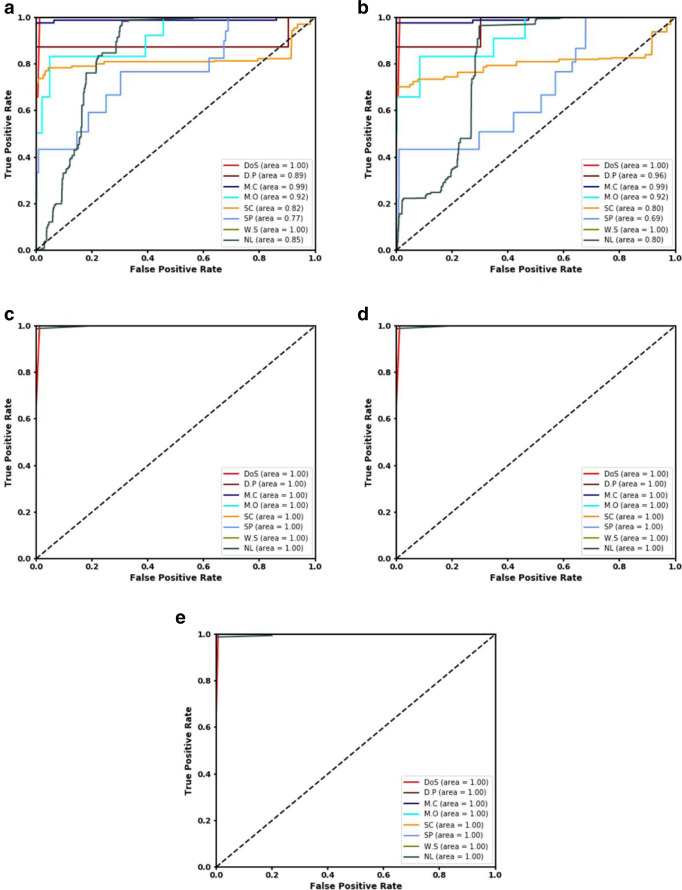




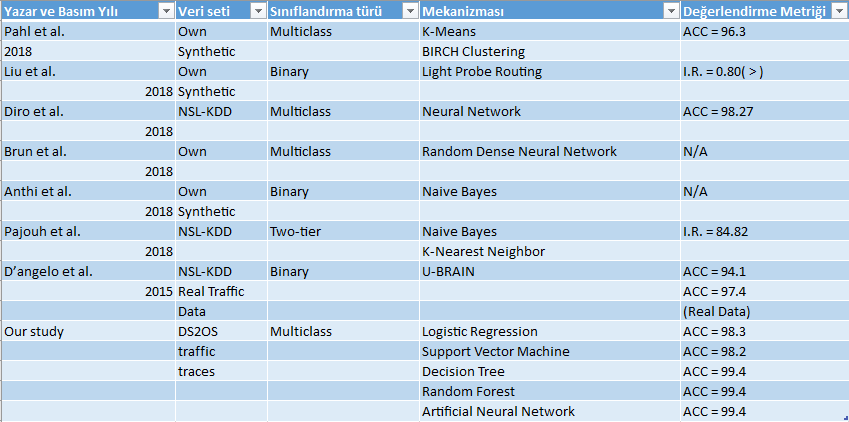


Şekil 7. KNeighbors, AdaBoost, GradientBoosting ,MLP, CatBoostClassifier, GaussianNB, DecisionTree, XGB, LogisticRegression, RandomForest’in Karışıklık Matrisi.

Son olarak, Şekil 8 sırasıyla LR, XGB, DT, RF ve ANN'nin Alıcı Çalışma Karakteristiği (ROC) Eğrilerini göstermektedir. Eğrilerin altındaki alandan, DT, XGB ve GradientBoosting'in daha yüksek doğruluğa sahip olduğu tanımlanabilir, çünkü her sınıf için eğrilerin altındaki alanların tümü yaklaşık olarak bir değere eşittir.



Şekil 8. (a) Lojistik Regresyon (b) XGB (c) Karar Ağacı (d) Rastgele Orman (e) Yapay Sinir Ağı.



Şekil 9. IoT saldırı tespitinin önceden yapılan çalışmalar ile karşılaştırılması.

Diğer makalelerle karşılaştırıldığında, makalemiz veri kümesinin çok ayrıntılı açıklamasını sağlar. Ayrıca, veri kümesi ön işleme adımlarının net bir açıklamasını sağlar. Makale, ikili sınıflandırmadan daha zor olan çoklu sınıfları sınıflandırmaya odaklanmıştır. Son olarak, bu yazıda her sınıflandırıcı için değerlendirme metrik değerlerinin net bir açıklaması verilmiştir.

Conclusion

Tam çalışmaya dayanarak, IoT ağındaki siber saldırıları çözmek için bu tür veri setlerinde XGB tekniğinin kullanılması gerektiği bulundu çünkü XGB, diğer yaklaşımlara kıyasla D.P, M.C, M.O, SC, SP, WS saldırılarını doğru bir şekilde öngördü. DoS ve Normal durumunda, diğer tekniklerden daha fazla numuneyi doğru tahmin etti. Dolayısıyla, bu tahminlere dayanarak, XGB'nin bu özel çalışma için en iyi teknik olduğu sonucuna varılabilir. Ancak burada veri kümesi üzerinde sadece klasik makine öğrenimi yaklaşımları kullanılmış ve karşılaştırmalı çalışma verilmiştir. Bu veri kümesinde yeni bir algoritma tasarlanmamıştır. Bu nedenle, sağlam bir algılama algoritması geliştirmek için daha fazla çalışmaya ihtiyaç vardır. Tüm çerçeve tasarımı üzerinde daha fazla analiz yapılmalıdır. Ayrıca bu çalışma sanal ortam verilerine dayanmaktadır. Gerçek zamanlı veriler söz konusu olduğunda, farklı sorunlar ortaya çıkabilir. Gerçek zamanlı verilere odaklanan bu problem üzerinde daha deneysel bir çalışmaya ihtiyaç vardır. IoT ağında, mikro hizmetler farklı zamanlarda farklı davranır ve bu da IoT hizmetlerinde normal davranışta sapmalara neden olarak bir anomali yaratır. Bu sorunları daha derinlemesine yorumlamak için daha fazla çalışmaya ihtiyaç vardır. Bu çalışmada, XGB yüksek doğruluk oranıyla nispeten daha iyi performans göstermektedir. Ancak, büyük veri ve diğer bilinmeyen sorunlar durumunda XGB'nin bu şekilde çalışacağını garanti etmez. Bu nedenle, daha fazla çalışmaya ihtiyaç duyulacaktır.

Future Work

Bu çalışma gelecekteki iyileştirmelere açıktır. Bu bölümde, bu iyileştirme olanaklarından birkaçı sunulacaktır.

Bu çalışmada eğitim ve test verisi olarak ağ akışından elde edilen özellikleri içeren CSV dosyalarından oluşan bir veri seti kullanılmıştır. Ne yazık ki, bu yöntem gerçek sistemlerde pratik olarak geçerli değildir. Ancak gerçek ağ verilerini yakalayan ve makine öğrenmesi algoritması ile çalışır hale getiren bir modül eklenerek bu sorun çözülebilir.

Diğer bir nokta ise bu çalışma sırasında çeşitli makine öğrenmesi yöntemlerinin birbirinden bağımsız olarak uygulanmış ve deneysel sonuçlar elde edilmiş olmasıdır. Ancak, bu yöntemin gerçek hayatta zayıf bir pratik uygulanabilirliği vardır. Bu sorunu çözmek için hiyerarşik bir makine öğrenimi yapısı tasarlanabilir. Üstelik böyle bir yapı sayesinde zamandan, CPU gücünden ve bellekten tasarruf etmek mümkün. Örneğin, iki katmanlı bir yapıda, ilk katman, Naive Bayes veya QDA gibi hızlı ve hesaplama açısından ucuz algoritmalardan oluşturulabilir, böylece ağ trafiği sürekli ve minimum maliyetle gözlemlenebilir. İlk adım, herhangi bir anomaliyi tespit ederken, bunu daha yüksek performans seviyesine sahip algoritmalardan oluşan bir üst katmana iletir. Bu katman karar mekanizmasını oluşturarak ölçüm yapar. İlk adım, bir anomaliyi tespit ederken, bunu ID3, AdaBoost ve KNN gibi daha yüksek performansa sahip algoritmalardan oluşan bir üst katmana iletir. Belirleme mekanizmasını oluşturan son katman, ağı saldırılara karşı korumak için ihtiyati karar alır.

References

[1] Johson, J. (2021). "*Worldwide digital population as of January 2021*", Statista. Retrieved from

<https://www.statista.com/statistics/617136/digital-population-worldwide/>

[2] Burhan, M., Rehman, R. A., Kim, B., Khan, B. (2018). "*IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey*", Department of Computer Science, National University of Computer and Emerging Sciences. Retrieved from

<https://www.researchgate.net/figure/Number-of-connected-IoT-devices-from-2012-to-2020_fig2_327272757>

[3] Unit 42. (2020). "*2020 Unit 42 IoT Threat Report*", Palo Alto Networks: Global Cybersecurity Leader. Retrieved from

<https://unit42.paloaltonetworks.com/iot-threat-report-2020/>

[4] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. (2019). "*Survey of intrusion detection systems: techniques, datasets and challenges*", Internet Commerce Security Laboratory, Federation University Australia. Retrieved from

<https://www.researchgate.net/publication/334533397_Survey_of_intrusion_detection_systems_techniques_datasets_and_challenges>

[5] Cobb, M., Loshin, P. (2018). "SSL (Secure Sockets Layer)", TechTarget. Retrieved from

<https://searchsecurity.techtarget.com/definition/Secure-Sockets-Layer-SSL>

[6] Guercio, K. (2021). *"Best Intrusion Detection and Prevention Systems for 2021: Guide to IDPS"*, eSecurity Planet: Internet Security for IT Professionals. Retrieved from

<https://www.esecurityplanet.com/products/intrusion-detection-and-prevention-systems/>

[7] FrancoisXA. (2018). “DS2OS traffic traces” Retrieved from https://www.kaggle.com/francoisxa/ds2ostraffictraces

[8] Ayon Dey. (2016). ”*Machine Learning Algorithms: A Review*”, Gautam Buddha University. Retrieved from http://ijcsit.com/docs/Volume%207/vol7issue3/ijcsit2016070332.pdf

[9] Fazakis, N. Kostopoulos, G. Karlos, S. Kotsiantis, S. Sgarbas K. (2019). "Self-trained eXtreme Gradient Boosting Trees", IEEE. Retrieved from https://ieeexplore.ieee.org/document/8900737

[10] Hancock, J. T. Khoshgoftaar, T. M. (2020). "CatBoost for big data: an interdisciplinary review", Journal of Big Data.

[11] An, T., Kim, M. (2010). *"A New Diverse AdaBoost Classifier"*, IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/5656396/

[12] Carolina Monard, M. (2002). "A Study of K-Nearest Neighbour as an Imputation Method", University of São Paulo. Retrieved from https://www.researchgate.net/profile/Maria-Carolina-Monard/publication/2475229\_A\_Study\_of\_K-Nearest\_Neighbour\_as\_an\_Imputation\_Method/links/0deec51ae1802c9861000000/A-Study-of-K-Nearest-Neighbour-as-an-Imputation-Method.pdf

[13] S.L. Ting, W.H. Ip, Albert H.C. Tsang. (2011). "Is Naïve Bayes a Good Classifier for Document Classification?", University, Hung Hum, Kowloon, Hong Kong. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.643.6611&rep=rep1&type=pdf

[14] David W. Hosmer, D. W. Lemeshow, J. S. (2000). Applied Logistic Regression. The Ohio State University. Retrieved from http://resource.heartonline.cn/20150528/1\_3kOQSTg.pdf

[15] Hernández, V. A. S. Monroy , R. Medina-Pérez , M. A. Loyola-González , O. Herrera, F. (2021). "A Practical Tutorial for Decision Tree Induction: Evaluation Measures for Candidate Splits and Opportunities". Retrieved from https://dl.acm.org/doi/abs/10.1145/3429739

[16] Hasan, Md. A. M.i Nasser, M. Ahmad, S. Molla, K. I. (2016). "Feature Selection for Intrusion Detection Using Random Forest". University of Rajshahi. Retrieved from https://scirp.org/journal/PaperInformation.aspx?PaperID=65359

[17] Zupan, J. (1994*)."Introduction to Artificial Neural Network (ANN) Methods: What They Are and How to Use Them"*, National Institute of Chemistry. Retrieved from https://www.researchgate.net/profile/Jure-Zupan-3/publication/251626579\_Introduction\_to\_Artificial\_Neural\_Network\_ANN\_Methods\_What\_They\_Are\_and\_How\_to\_Use\_Them/links/541df0e70cf241a65a189e66/Introduction-to-Artificial-Neural-Network-ANN-Methods-What-They-Are-and-How-to-Use-Them.pdf

[18] Visa, S. Ramsay, B. Ralescu A. Knaap E. V. D. (2011). "Proceedings of the Twenty­second Midwest Artificial Intelligence and Cognitive Science Conference", University of Cincinnati. Retrieved from https://www.researchgate.net/profile/Atsushi-Inoue-6/publication/220833227\_Page\_Ranking\_Refinement\_Using\_Fuzzy\_Sets\_and\_Logic/links/54b743480cf24eb34f6e9e80/Page-Ranking-Refinement-Using-Fuzzy-Sets-and-Logic.pdf#page=126

[15] M. Junker; R. Hoch; A. Dengel. (1999). "On the evaluation of document analysis components by recall, precision, and accuracy", IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/791887

[16] Hashemi, S. Anthony, N. Tann, H. Bahar, R.I. Reda, S. (2017). "Understanding the impact of precision quantization on the accuracy and energy of neural networks", IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/7927224

[17] Juba, B. Le, Hai S. (2019). "Precision-Recall versus Accuracy and the Role of Large Data Sets", Washington University. Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/5193

[18] Chicco, D. Jurman, G. (2020). "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation", BMC Genomics. Retrieved from https://link.springer.com/article/10.1186/s12864-019-6413-7

[19] Gunawardana, A. Shani, G. (2009). "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks", Ben Gurion University. Retrieved from https://www.jmlr.org/papers/volume10/gunawardana09a/gunawardana09a.pdf

[20] Lemenkova, P. (2019)."PROCESSING OCEANOGRAPHIC DATA BY PYTHON LIBRARIES NUMPY, SCIPY AND PANDAS", Ocean University of China. Retrieved from http://aquatres.scientificwebjournals.com/en/download/article-file/690837