KARADENİZ TEKNİK ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Zeynep İlkay ŞAHİN – 394824

Süveyda CAN – 394792

Ayşenur TAK – 394774

Sevgi YILMAZ - 394813

Ağ Güvenliği

Prof. Dr. GÜZİN ULUTAŞ

Integrating Feature Selection with Voting Classifiers for Effective Intrusion Detection

İçindekiler

1.GİRİŞ

Bu projede çeşitli IDS (Saldırı Algılama Sistemleri) ile ilgili çalışmalar titizlikle incelenmiştir. Bu incelemeler, farklı metodolojilerin detaylı bir şekilde ele alınmasını ve bu metodolojilerin birleştirilerek elde edilen sonuçların analiz edilmesini hedeflemektedir. Ayrıca, mevcut algoritmalara yeni metotların entegrasyonu da göz önünde bulundurulmuştur. Bu çalışmada, IDS alanında yenilikçi yaklaşımların geliştirilmesi ve mevcut sistemlerin performansının artırılmasına yönelik bir çalışma gerçekleştirilmiştir.

Projenin genel içeriği şu şekildedir: DoS, Probe, U2R ve R2L saldırıları için özellik çıkarımı yapılmıştır. Bu özellikler, makine öğrenmesi algoritmalarıyla birleştirilerek saldırıların tespit edilmesinde ne kadar doğru tahminler yapılabileceği incelenmiştir. Son olarak, kullanılan tüm makine öğrenimi algoritmalarının tahmin sonuçları, Voting Karar Mekanizması kullanılarak birleştirilmiştir. Bu karar mekanizması sayesinde tüm modellerin katkısı değerlendirilerek nihai bir tahmin sonucu elde edilmiştir.

İncelenen makalelerden alınan yöntemler şu şekildedir: "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security" makalesinde makine öğrenmesi algoritmaları ve derin öğrenme modellerinin IDS sistemindeki başarı oranları karşılaştırılmıştır. Bu projede ise bahsi geçen makaleden alınan Logistic Regression, Gaussian, Decision Tree, AdaBoost ve Random Forest gibi makine öğrenmesi algoritmaları kullanılmıştır. Ayrıca, "A Subset Feature Elimination Mechanism for Intrusion Detection System" makalesinde, özellik çıkarımı yapılarak seçilen özelliklerin saldırı tespiti doğruluğuna etkisi incelenmiştir. Bu makalede, birden çok özellik çıkarım yöntemi ele alınmıştır; ancak projemizde ANOVA-F istatistiksel testi ile özellik çıkarım yöntemi kullanılmıştır.

2. LİTERATÜR TARAMASI

Kitsune IDS, Yisroel Mirsky, Tomer Doitshman, Yuval Elovici ve Asaf Shabtai tarafından geliştirilen, otoenkoderlere (autoencoder) dayalı yeni bir ağ saldırısı algılama sistemi (NIDS)dir. Hafif ve gözetimsiz doğasıyla öne çıkan Kitsune, etiketlenmiş veriye ihtiyaç duymadan ağ trafiği modellerini öğrenerek ve birden fazla küçük otoenkoderden oluşan bir topluluk kullanarak anormallikleri algılar. Bu sayede basit ağ cihazlarında bile konuşlandırılabilir ve geleneksel NIDS'lere kıyasla daha geniş bir yelpazede kullanılabilir. Kitsune IDS; özellik çıkarma, anormallik algılama ve karar verme olan üç ana aşamadan oluşur. Hafif, gözetimsiz, verimli ve gerçek zamanlı olması avantajları arasındadır. Yalnızca ağ trafiği anormalliklerini algılayabildiği için diğer saldırı türlerini (örn: kimlik avı) tespit edemeyebilir. [1]

"Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security" makalesi Rahul Vigneswaran K., Vinayakumar R., Soman KP. ve Prabaharan Poornachandran. tarafından geliştirilen siber güvenlikte ağ saldırı tespiti için kullanılan yüzeysel (shallow) ve derin (deep) sinir ağlarının değerlendirilmesini ele almaktadır. Ana hedef, Derin Sinir Ağlarının (DNN'ler) ağ saldırı tespit sistemlerinde (N-IDS) saldırıları tahmin etme yeteneklerini incelemektir.

KDDCup-’99 veri seti kullanılarak çeşitli klasik makine öğrenme algoritmaları ve farklı katman sayısına sahip DNN'ler (1'den 5'e kadar) eğitilmiş ve karşılaştırılmıştır. Sonuçlar, 3 katmanlı bir DNN'nin diğer klasik makine öğrenme algoritmalarına kıyasla üstün performans gösterdiğini ortaya koymuştur. DNN'lerin, özellikle ReLU (Rectified Linear Unit) aktivasyon fonksiyonunun uygulanması ile birlikte, eğitim sürecini hızlandırdığı ve vanishing gradient problemine çözüm sunduğu vurgulanmıştır. Derin sinir ağlarının IDS'lerle birleştirilmesinin, ağ saldırılarını insan gözüyle tespit edilemeyecek düzeyde tespit edebileceği ve otomatik yanıtlar üretebileceği sonucuna varılmıştır. [2]

“A Subset Feature Elimination Mechanism for Intrusion Detection System” makalesi Malezya Teknoloji Üniversitesi (Universiti Teknologi Malaysia) Bilgisayar Fakültesi'nden Herve Nkiama, Syed Zainudeen Mohd Said ve Muhammad Saidu tarafından geliştirilen IDS'nin verimliliğini ve etkinliğini artırmada önemli bir adım olan özellik seçimini ele almaktadır. Bu çalışmada, karar ağacı sınıflandırıcı kullanarak özyinelemeli özellik eleme (RFE) ile birleştirilen tek değişkenli özellik seçimini içeren yeni bir özellik seçme mekanizması tanıtılmaktadır. Bu yöntem, her bir özelliğin önemini sistematik olarak değerlendirir ve en az önemli özellikleri yinelemeli olarak çıkararak ve modeli yeniden inşa ederek optimal özellik alt kümesi belirlenene kadar devam eder. Bu yaklaşım, KDD 1999 veri setinin rafine edilmiş bir versiyonu olan NSL-KDD veri setine uygulanmış ve hem doğrulukta hem de işleme hızında önemli iyileşmeler göstermiştir. Sonuçlar, özellik sayısını azaltmanın sadece modelin doğruluğunu artırmakla kalmayıp, aynı zamanda sınıflandırıcıyı oluşturma süresini de azalttığını göstermiştir. Örneğin, R2L (Remote to Local) saldırılarını tespit etme doğruluğu %97.03'ten %99.88'e yükselirken, DoS (Denial of Service) saldırıları için sınıflandırıcıyı oluşturma süresi 15.5 saniyeden 0.959 saniyeye düştüğü görülmüştür.

Makalede ANOVA yöntemi, özellik seçimi sürecinde kullanılan bir yöntem olarak belirtilmiştir. Özellik seçimi, gereksiz ve alakasız verileri elemek için kullanılan bir tekniktir. Bu süreçte, özelliklerin her birinin hedef sınıflarla olan ilişkisini belirlemek için univariate özellik seçimi ve ANOVA F-testi kullanılmıştır. [3]

"An Intrusion Detection System based on Deep Belief Networks" makalesi Othmane Belarbi, Aftab Khan, Pietro Carnelli ve Theodoros Spyridopoulos tarafından yazılmıştır. Makalede, Deep Belief Networks (DBN) tabanlı bir Network Intrusion Detection System (NIDS) önerilmektedir. Bu yöntem, veri dengesizliğinin üstesinden gelmek ve yüksek başarımlı saldırı tespiti sağlamak için derin öğrenme tekniklerini kullanmaktadır. DBN, birden fazla Restricted Boltzmann Machine (RBM) katmanının istiflenmesiyle oluşturulan bir üretken grafik modelidir. DBN'nin eğitim süreci iki aşamadan oluşur: öncelikle katman katman açgözlü bir öğrenme algoritması ile denetimsiz bir şekilde ön eğitim yapılır, ardından geri yayılım tekniği ile denetimli bir şekilde ince ayar yapılır. Makale, CICIDS2017 veri setini kullanarak DBN tabanlı NIDS modelinin performansını değerlendirmiştir. Araştırmada çeşitli sınıf dengeleme teknikleri uygulanmış ve bu tekniklerin performans üzerindeki etkileri incelenmiştir. Özellikle, SMOTE ve rastgele alt örnekleme teknikleri kullanılarak DBN ve MLP tabanlı NIDS modellerinin F1-skorları değerlendirilmiştir. Deneysel sonuçlar, DBN tabanlı modelin, özellikle veri setinde az sayıda örneği bulunan saldırılar üzerinde MLP tabanlı modele göre daha iyi performans gösterdiğini ortaya koymuştur. [4]

2.1.Makine Öğrenmesi ve Ağ Güvenliği

Saldırı Algılama Sistemleri (IDS) projelerinde kullanılan makine öğrenimi algoritmaları bir dizi avantaja sahiptir. Öncelikle, bu algoritmalar veriye dayalı olarak öğrenir ve zamanla deneyimlerine göre iyileşirler. Genellikle büyük miktarda veriyle başa çıkabilirler ve karmaşık ilişkileri öğrenebilirler, bu da geleneksel yöntemlerden daha etkili sonuçlar elde etmelerini sağlar. Otomatik olarak öğrenme ve adapte olma yetenekleri, hızlı değişen tehditlere karşı daha uygun hale gelmelerini sağlar. Bu nedenle, IDS projelerinde makine öğrenimi algoritmaları sıklıkla tercih edilir ve genellikle verimlilik ve doğruluk açısından geleneksel yöntemlere göre avantaj sağlarlar.

Ayrıca, Feature Elimination Mechanism yönteminin kullanılması da birçok avantaja sahiptir. Bu yöntem, özellik seçiminde kullanılarak, model performansını artırmak ve hesaplama kaynaklarını etkin bir şekilde kullanmak için gereksiz veya etkisiz özellikleri belirleme ve kaldırma işlemini ifade eder. Gereksiz özelliklerin çıkarılmasıyla birlikte, model daha az karmaşık hale gelir ve bu da genellikle daha iyi genelleme yeteneği sağlar. Ayrıca, gereksiz özelliklerin kaldırılması, modelin daha anlaşılır ve yorumlanabilir olmasını sağlar, böylece hangi özelliklerin önemli olduğunu daha net bir şekilde görebiliriz.

3.PROJE TASARIMI VE METODOLOJİ

3.1.Yöntem

3.1.1.Kullanılan Yöntemlerin Avantajları ve Seçim Nedenleri

Projemizde ağ güvenliği tehditlerini tespit etmek için çeşitli makine öğrenmesi algoritmalarını kullandık. Her bir algoritmanın kendine özgü avantajları bulunmaktadır ve projemizde bu algoritmaları seçme nedenlerimiz şu şekildedir:

- Lojistik Regresyon (Logistic Regression – LR)

Lojistik regresyon, projemizde temel bir sınıflandırma yöntemi olarak kullanılmıştır. Ağ güvenliği alanında temel tehditlerin belirlenmesi için iyi bir başlangıç noktası sağlar ve diğer daha karmaşık modellerle karşılaştırma yapmamıza olanak sağlar. Çıktıkların olasılıkları olarak yorumlanabilmesi nedeniyle oldukça anlaşılır ve yorumlanabilir bir modeldir. Eğitim süreci hızlıdır ve büyük veri setlerinde bile verimli çalışabilir. Ayrıca basit yapısı sayesinde aşırı uyum (overfitting) riski düşüktür.

- Naive Bayes (NB)

Naive Bayes, özellikle veri setinde bağımsız değişkenlerin varsayıldığı durumlarda etkili bir sınıflandırıcıdır. Büyük veri setlerinde dahi hızlı bir şekilde eğitilebilir. Büyük veri setlerinde olduğu gibi küçük veri setlerinde de iyi performans gösterir. Algoritmanın basit ve hızlı olması uygulamada kolaylık sağlar. Ağ güvenliği tehditlerinin hızlı ve etkili bir şekilde tespit edilmesi için kullanılabilir.

- Karar Ağaçları (Decision Trees –DT)

Karar ağaçları, ağ güvenliği tehditlerinin belirlenmesinde karar kurallarının anlaşılabilir olması nedeniyle tercih edilmiştir. Bu, tehditlerin neden ve nasıl tespit edildiğini açıkça gösterilebilir. Karar ağaçları, karar kuralları ve görselleştirme sayesinde kolayca yorumlanabilir. Hedefe yönelik değişken seçimi konusunda, önemli özelliklerin belirlenmesinde yardımcı olur. Veri ön işleme ve normalizasyon gereksinimi düşüktür.

- AdaBoost (Adaptive Boosting)

AdaBosst, ağ güvenliği tehditlerinin tespitinde yüksek performans sunar. Farklı zayıf öğrenicileri birleştirerek daha doğru ve güvenilir sonuçlar elde etmemizi sağlar. Zayıf öğrenicileri birleştirerek güçlü bir öğrenici oluşturur ve genellikle yüksek doğruluk sağlar. Aşırı uyum (Overfitting) riskini azaltır. Esneklik kabiliyeti yüksektir. Yani çeşitli zayıf öğrenicilerle kullanılabilir.

- Random Forest (RF)

Random Forest, karmaşık veri setlerinde bile yüksek doğruluk sağlar. Ağ güvenliği tehditlerinin tespitinde güvenilir ve genelleştirilebilir sonuçlar elde etmemize olanak tanır. Birçok karar ağacının birleştirilmesiyle elde edilen tahminler genellikle yüksek doğruluk sağlar. Overfitting riskini azaltır ve genelleme yeteneğini artırır. Ayrıca hangi özelliklerin daha önemli olduğunu belirlemeye yardımcı olabilir.

Projemizde bu yöntemleri seçmemizin temel nedeni, her birinin ağ güvenliği tehditlerinin tespitinde farklı avantajlar sunmasıdır. Lojistik regresyonun basit ve anlaşılır yapısı, Naive Bayes'in hız ve verimliliği, karar ağaçlarının yorumlanabilirliği, AdaBoost'un yüksek performansı ve Random Forest'ın doğruluk ve genelleme yeteneği gibi özellikler, projemizin kapsamını genişletmiş ve tespit doğruluğunu artırmıştır. Bu yöntemlerin kombinasyonu, ağ güvenliği tehditlerini etkili bir şekilde belirlememize yardımcı olmuş ve projemizin başarısını sağlamıştır.

Projemizde kullanılan Anova (Analysis of Variance) yöntemi, özellikle özellik seçimi (feature selection) sürecinde önemli bir rol oynamaktadır. Anova'nın avantajları ve bu projede kullanılma nedenleri şu şekildedir:

- Özelliklerin Önemini Belirleme: Anova, bağımsız değişkenlerin bağımlı değişken üzerindeki etkisini ölçerek hangi özelliklerin model için daha önemli olduğunu belirler. Bu, modelin doğruluğunu artırmak ve gereksiz özellikleri elimine etmek için faydalıdır.
- Basit ve Hızlı: Hesaplama açısından basit ve hızlıdır. Büyük veri setlerinde bile kolayca uygulanabilir.
- İstatistiksel Anlamlılık: Anova, her bir özelliğin bağımlı değişken üzerindeki etkisinin istatistiksel olarak anlamlı olup olmadığını belirler. Bu, modelin güvenilirliğini artırır.
- Boyut Azaltma: Özellik seçimi ile veri boyutunu azaltarak daha basit ve hızlı modeller oluşturulmasını sağlar. Bu, özellikle yüksek boyutlu veri setlerinde modelin eğitim süresini ve performansını olumlu yönde etkiler.

Projemizde Anova'yı tercih etmemizin başlıca nedeni, ağ güvenliği tehditlerinin tespitinde kullanılan özelliklerin önemli olup olmadığını belirlemek ve modelin performansını artırmaktır. Anova, özellikle büyük ve karmaşık veri setlerinde, önemli özelliklerin seçilmesini ve gereksiz özelliklerin elimine edilmesini sağlar. Bu, modelin daha doğru ve güvenilir sonuçlar üretmesine yardımcı olur.

ANOVA ve RFE Karşılaştırması:

- ANOVA: Anova, her bir özelliğin bağımlı değişken üzerindeki etkisini bağımsız olarak değerlendirir ve istatistiksel anlamlılık testi yapar. Özellikle bağımsız değişkenler arasında korelasyonun düşük olduğu durumlarda etkilidir.
- RFE (Recursive Feature Elimination): RFE, özellikleri iteratif olarak elimine eder ve modelin performansına göre en iyi özellik setini belirler. Bu yöntem daha karmaşık ve hesaplama açısından daha yoğundur, ancak daha hassas ve detaylı bir özellik seçimi sağlar.

Anova'nın bu projede tercih edilmesinin nedeni, ağ güvenliği tehditlerinin tespitinde hızlı ve etkili bir özellik seçimi sağlamasıdır. Anova, özellikle yüksek boyutlu veri setlerinde hızlı ve güvenilir sonuçlar sunar. Ayrıca, her bir özelliğin istatistiksel anlamlılığını belirleyerek

modelin doğruluğunu artırır. Bu nedenle, Anova projemizde ağ güvenliği tehditlerini etkili bir şekilde belirlemek için ideal bir seçimdir.

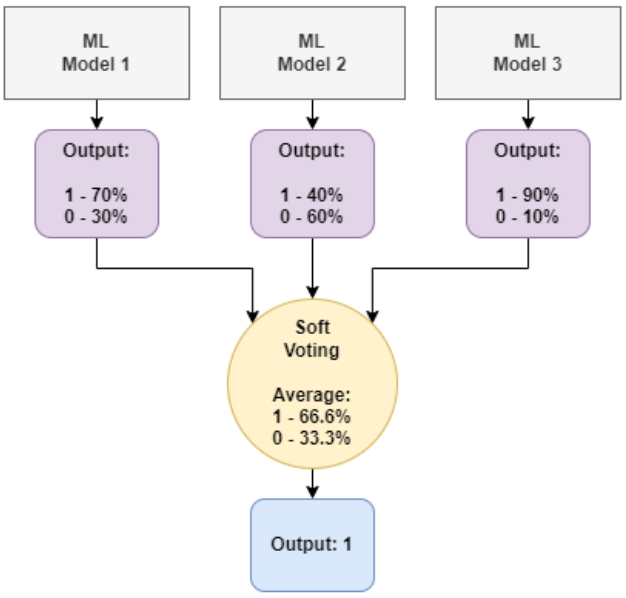
Voting ve AdaBoost karşılaştırması:

AdaBoost (Adaptive Boosting) algoritması, zayıf sınıflandırıcıları ardışık olarak birleştirerek güçlü bir sınıflandırıcı oluşturan bir güçlendirme tekniğidir.

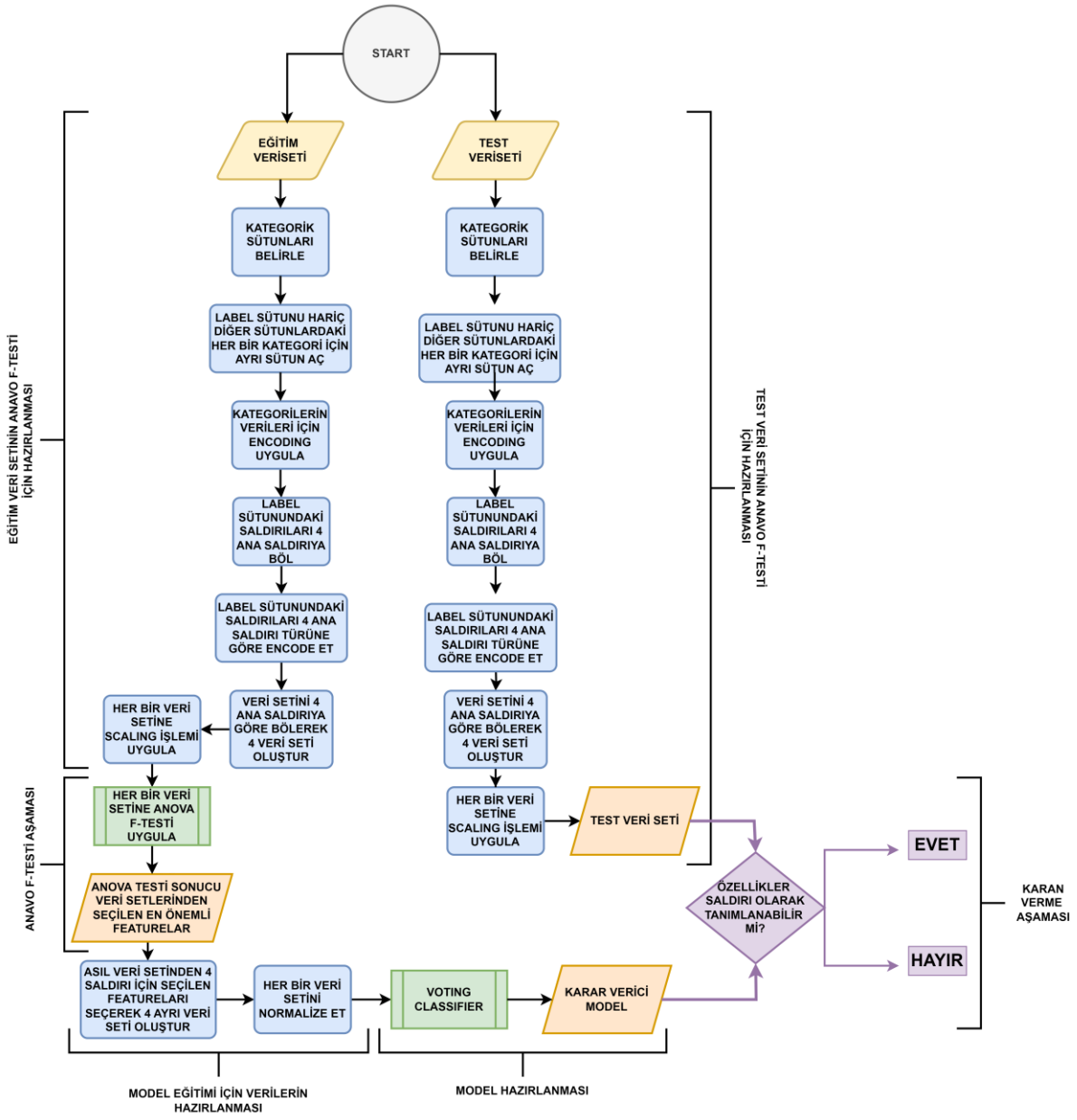
AdaBoost, hataları düzelterek ve sınıflandırıcıları ağırlıklandırarak daha yüksek doğruluk sağlayabilir, ancak gürültülü verilere karşı hassastır ve hesaplama maliyeti yüksektir. Voting ise daha basit, uygulanması kolay ve gürültüye karşı daha dayanıklıdır, ancak bireysel hataları düzeltmez ve performansı, kullanılan sınıflandırıcıların kalitesine bağlıdır. AdaBoost yöntemi donanım maliyetinin yüksekliği sebebiyle tercih edilmemiştir.

Projede karar mekanizması olarak Voting sınıflandırma kullanılmıştır. Bu yöntemin kullanılmasının sebebi ensemble öğrenme tekniklerinden biri olması olup, birden fazla sınıflandırma veya regresyon modelini bir araya getirerek daha güçlü ve dengeli bir tahmin yapma yeteneği sağlamasıdır.Hard voting ve Soft voting olmak üzere iki alt yöntem bulunmaktadır. Hard voting, en basit topluluk öğrenme yöntemidir. Her sınıflandırıcı bir tahminde bulunur ve çoğunluğun tahmini, topluluğun kararı olur. Örneğin, üç sınıflandırıcı bir görüntüyü kedi olarak tanımlarken, bir sınıflandırıcı geyiğe ait olduğunu belirtiyorsa, topluluğun tahmini kedi yönünde olur. Soft voting ise sınıflandırıcıların tahminlerinin güvenilirliğini dikkate alır. Her sınıflandırıcı, her sınıfa bir olasılık atar ve en yüksek toplam olasılığa sahip sınıf, topluluğun tahmini olur. Örneğin, bir sınıflandırıcı bir görüntünün kedi olduğunu %90 olasılıkla, bir diğeri geyiğe ait olduğunu %10 olasılıkla tahmin ediyorsa, topluluğun kararı kedi olacaktır. Uygulamamızda Soft voting yöntemi tercih edilmiştir.

Bu yöntem, farklı özelliklere veya algoritmaların birleştirilmesiyle çeşitlilikten faydalanır ve her bir modelin kendi hatalarını telafi etmesini sağlar. Sonuç olarak, daha iyi genelleme yeteneği, yüksek kesinlik ve daha az aşırı uyum gibi avantajlar sunar. [5]



4. AKIŞ DİYAGRAMI VE AÇIKLAMASI



Resim 1.1

Projede geliştirdiğimiz sistemin akış diyagramı Resim 1.1 de görüldüğü gibidir.

Sistemin çalışması şu 6 adımda incelenebilir; eğitim veri setinin ANOVA F-Testi için hazırlanması, test veri setinin ANOVA F-Testi için hazırlanması, ANOVA F-Testi uygulanması, model eğitimi için verilerin hazırlanması, karar verici modelin yapısının hazırlanması, karar verme aşaması.

4.1. Eğitim Verisetinin ANOVA F-Testi için hazırlanması

İlk aşama olan eğitim veri seti hazırlanmalıdır. Projemizde NSL KDD veri seti kullanılmıştır. Veri seti sütunların isimlerini belirterek .csv dosyası olarak projeye dahil edilmelidir.

Veri setinde bazı kategorik verilere sahip sütunlar bulunmaktadır. Bu kategorik sütunlar belirlenip ilgili işlemler uygulanmalıdır. Kategorik verilere sahip sütunlar *protocol_type, service, flag,* ve *label* sütunlarıdır. Label sütunu hariç diğer sütunlardaki her bir kategori belirlenir ve bu kategoriler *dummy columns* olarak veri setine ayrı sütunlar halinde eklenir. Kategorik sütun verilerine label encoding işlemi yapılır. Bu işlemin çıktısı yardımıyla *dummy columns* olarak hazırlanan sütunların verilerine One Hot Encoding işlemi uygulanır.

label sütununda ilgili satırın hangi saldırıya ait olduğu bilgisi bulunmaktadır. *label* satırındaki her bir saldırı kategorisi belirlenir. Bu saldırı türleri DoS, Probe, R2L, U2R saldırıları olarak ayrıştırılırlar. Yenilenmiş *label* sütunu ile veri seti bu seferde bu 4 ana saldırı kategorisi için bölünür. Örneğin veri setindeki DoS saldırısına dahil olan satırlar DoS veri setine alınır. Son durumda 4 farklı saldırı için ayrı ayrı veri setleri elde edilir. Bu aşamanın son kısmında ise bu saldırılara göre ayrılmış veri setleri scaling edilerek ANOVA F-Testine hazır hale getirilir.

4.2. Test Verisetinin ANOVA F-Testi için hazırlanması

Test veri seti için preprocessing kısmının eğitim veri setine uygulananlardan tek farkı test veri setinde eğitim veri setine göre daha az kategori bulunmuştur. Bu yüzden test veri setine eksik kategoriler eklenir.

4.3. ANOVA F-Testi uygulanması

ANOVA F-testi aşamasında *sklearn.feature_selection* kütüphanesinden *SelectPercentile* ve *f_classif* modülleri kullanılmıştır. 4 saldırı veri seti içinde bu test uygulanır ve ilgili veri setleri içinde en anlamlı sütunlar-özellikler elde edilir.

4.4. Model eğitimi için verilerin hazırlanması

Bu aşamada saldırı tespiti yapacak modeller için veri setleri tekrar düzenlenir. Her bir veri setinden, seçilen featureların olduğu sütunlar ayırt edilir ve en anlamlı özelliklerden oluşan yeni saldırı veri setleri hazırlanır. Her bir veri seti normalize edilir.

4.5. Karar verici modelin yapısının hazırlanması

Karar verici model için ensemble yöntemi olan Voting Classifier içinde Logistic Regression, Gaussian Naive Bayes, Decision Tree Classifier, Ada Boost Classifier, Random Forest Classifier sınıflandırma algoritmaları kullanılmıştır. Bu oylama yoluyla sınıflandırma tekniği sayesinde bir çok algoritmayı birleştirerek daha doğru kararlar verebilecek bir karar verici model oluşturulmuştur.

Voting Classifier modelleri, her bir ana özellikleri seçilmiş saldırı veri seti ile eğitilir ve 4 saldırı için 4 model oluşturulur.

4.6. Karar verme aşaması

Karar verme aşamasında test verileri Voting Classifier nesnesine verilerek tahminler elde edilir.

5. VERİ SETİ VE ÖZELLİKLERİN KULLANIMI

KDD Cup 1999 veri seti, ağ saldırı tespiti ve ağ güvenliği alanında yaygın olarak kullanılan bir benchmark veri setidir. Bu veri setleri ve özellikleri (features) kullanmamızın nedenleri aşağıdaki gibi açıklanabilir:

Veri Seti ve Özelliklerin Kullanım Nedenleri

- Güvenilir ve Tanınmış: KDD Cup 1999 veri seti, ağ güvenliği araştırmalarında ve saldırı tespiti sistemlerinde yaygın olarak kullanılan, güvenilir bir veri setidir. Bu veri seti, literatürde sıkça referans alınır ve bu nedenle yapılan çalışmaların sonuçlarının karşılaştırılabilirliği artar.
- Kapsamlı: Veri seti, farklı türlerde birçok saldırıyı ve normal trafik kayıtlarını içerir. Bu çeşitlilik, modellerin geniş bir yelpazede tehditleri tespit etmesine olanak tanır.
- Detaylı ve Çeşitli Özellikler: Veri setinde 41 farklı özellik bulunmaktadır. Bu özellikler, ağ trafiğinin detaylı ve çeşitli yönlerini kapsar. Saldırı ve Normal Trafik Ayrımı: Veri seti, saldırı türlerini ve normal ağ trafiğini açıkça ayırır. Bu, makine öğrenmesi algoritmalarının eğitim ve test süreçlerinde saldırıları ve normal trafiği ayırt etmesini sağlar.
- Kapsamlı Etiketleme: Veri seti, her bir bağlantı kaydının saldırı veya normal trafik olup olmadığını belirten etiketler içerir. Bu, denetimli öğrenme algoritmalarının etkili bir şekilde eğitim almasını sağlar.
- Büyük ve Dengeli Veri Seti: Veri setinin boyutu, makine öğrenmesi algoritmalarının etkin bir şekilde eğitilmesi için yeterlidir. Ayrıca, veri seti, farklı saldırı türlerinin dengeli bir dağılımını içerir, bu da algoritmaların farklı saldırı türlerini öğrenmesini ve tespit etmesini sağlar.

Özelliklerin (Features) Detaylı İncelemesi

- Temel Ağ Trafiği Bilgileri: duration, protocol_type, service, flag, src_bytes, dst_bytes gibi özellikler, temel ağ trafiği bilgilerini içerir ve hangi protokollerin, hizmetlerin ve bayrakların kullanıldığını belirtir.
- Oturum ve Giriş Bilgileri: num_failed_logins, logged_in, is_host_login, is_guest_login gibi özellikler, oturum açma ve kullanıcı bilgileriyle ilgilidir.
- Bağlantı İstatistikleri: count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate gibi özellikler, belirli bir zaman diliminde veya hizmetteki bağlantı sayısını ve hata oranlarını belirtir.
- Ağ Davranışları ve Anomalileri: hot, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds gibi özellikler, ağdaki anormal aktiviteleri ve olası saldırı göstergelerini belirtir.
- Host ve Hizmet İstatistikleri: dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate,

dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate gibi özellikler, hedef host ve hizmetlerin istatistiklerini içerir.

Bu nedenlerle, KDD Cup 1999 veri seti ve özellikleri, ağ güvenliği projeleri için ideal bir seçimdir. Özelliklerin çeşitliliği ve veri setinin kapsamlı yapısı, ağ güvenliği tehditlerini etkili bir şekilde tespit etmek ve değerlendirmek için gerekli tüm bilgileri sağlar.

6. DENEY VE TEST AŞAMASI

Veri setlerinin ön işleme aşaması "Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security" çalışmasından değiştirilmeden alınmıştır. "A Subset Feature Elimination Mechanism for Intrusion Detection System" çalışmasından ise ANNOVA F-Testi uygulaması değiştirilmeden alınmıştır. Bizim çalışmamızın deney kapsamı ise özellik seçimi (feature selection) sonuçları ile makine öğrenimi modellerini Voting Model üzerinde birleştirilmesinin nihai sonuca nasıl etki edeceği üzerinedir.

Projemizin geliştirilmesinde öncelikle makine öğrenmesi modellerinde nasıl bir veri seti kullanacağımız üzerine çalışmalar yürüttük. Başlangıçta önişleme sonucu elde edilen *new_df* veri setini her saldırı özelinde seçilen özellikler ile sınırlayarak bir veri seti oluşturma yoluna gidilmiştir. Ayrıca model performans değerlendirmelerinde *average* değeri de *weighted* olarak kullanılmıştır.

DoS saldırı tipi için makine öğrenmesi modellerinin sonuçları şu şekilde gözlenmiştir,

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.626	0.592	0.626	0.560
Gaussian NB	0.404	0.698	0.404	0.482
Decision Tree Classifier	0.696	0.701	0.696	0.650
Adaboost	0.635	0.605	0.635	0.604
Random Forest Classifier	0.678	0.729	0.678	0.633
Voting Classifier	0.686	0.718	0.686	0.640

Yukarıda görülen değerlerin düşük oluşu nedeniyle ANNOVA F-Testinde de kullanılan saldırılar özelinde oluşturulmuş veri setleri ile bir deneme yapılmıştır. Sonuçlar aşağıdaki gibi elde edilmiştir.

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.818	0.830	0.818	0.813
Gaussian NB	0.813	0.814	0.813	0.811
Decision Tree Classifier	0.860	0.868	0.860	0.858
Adaboost	0.836	0.847	0.836	0.832
Random Forest Classifier	0.828	0.840	0.828	0.823
Voting Classifier	0.829	0.840	0.829	0.824

Sadece DoS saldırı değerlerini içeren bir veri seti kullanılıyorken ve çıkış etiketleri sadece 1 ve 0 iken (DoS ve normal değerleri için kullanılan etiketler) performans metriklerindeki *average* değeri *binary* olarak değiştirilmiş ve sonuçlar bir de bu şekilde incelenmiştir. Sonuçlar aşağıda gösterilmiştir.

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.818	0.890	0.664	0.760
Gaussian NB	0.813	0.825	0.724	0.771
Decision Tree Classifier	0.866	0.920	0.758	0.831
Adaboost	0.836	0.907	0.695	0.787
Random Forest Classifier	0.830	0.904	0.681	0.777
Voting Classifier	0.829	0.901	0.680	0.775

Bir önceki deney sonuçları ile çok büyük bir fark oluşmadığından sonuçlar bir de Probe üzerinde denenmiştir. Veri seti olarak Probe saldırısı özelinde hazırlanmış veri seti Probe için seçilmiş özellikler ile sınırlandırılmıştır. *Binary* parametresi iki etiket içeren veri setleri için uygun bir seçim olabilmektedir. Fakat bu aşamada *binary* kullanılması için veri setlerinde küçük bir değişikliğe gidilmiştir. Probe etiketleri 0 ve 2'den oluştuğu için *binary* parametresi kullanılamazdı dolayısıyla Probe saldırı etiketi olan 2 değerlerini Probe saldırı veri setinde 1 ile değiştirdik. Bu değişikliklerin ardından değerleri şu şekilde gözlemledik.

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.877	0.891	0.437	0.586
Gaussian NB	0.886	0.849	0.523	0.647
Decision Tree Classifier	0.883	0.841	0.510	0.635
Adaboost	0.892	0.850	0.558	0.674
Random Forest Classifier	0.888	0.847	0.537	0.658
Voting Classifier	0.890	0.855	0.539	0.661

Değerlerin beklenenden düşük çıkması bizleri veri setini daha iyi incelemeye itmiştir. Bu noktada veri setlerindeki etiket dağılımı incelenmiştir. Aşağıda her saldırı veri seti için etiket dağılımı tablolar halinde gösterilmiştir.

DoS

Label	Count
0	67343
1	45927

Probe

Label	Count
0	67343
2	11656

R2L

Label	Count
0	67343
3	995

U2R

Label	Count
0	67343
4	52

Veri setlerinde dengesizlik model sonuçlarında ciddi sorunlara yol açabilmektedir. Modelin sonuçları yanlı olabilmekte, azınlık sınıfın performansı düşük olabilmektedir. Genel olarak modelin doğruluk performansı, genelleme yeteneği düşük olacaktır. Dengesiz veri seti olduğu durumlarda çeşitli yaklaşımlar uygulanır. Bizler bu yaklaşıklardan bazılarını denedik ve sonuçlarını inceledik. Denemelerimizi normal etiket değeri yaklaşık 67 kat daha fazla olan R2L saldırısının veri seti ve modelleri üzerinde denedik.

İlk olarak Undersampling (Çoğunluk sınıfını azaltma) yöntemi denenmiştir. Bu denemelerde çoğunluk sınıf olan normal (0 etiketli sınıf) sınıfından saldırı sınıfının veri miktarının ½ kadarı ve ¾'ü kadarı ile denemeler yapılmıştır. Performans değerlendirmelerinde ise *average* değerini *weighted* olarak kullanılmıştır. Bu denemelerdeki R2L saldırı tipi için geliştirilen Voting Classifier Model sonuçları aşağıdaki gibi olmuştur.

R2L saldırı etiketlerinin ½ oranında normal örneklemesi sonucunda 1492 adet verinin 995 tanesi R2L saldırı, 497 tanesi normal etiketlidir. Buna göre model sonucu şu şekildedir:

	Accuracy	Precision	Recall	F1 Score
Voting Classifier	0.423	0.886	0.0.423	0.469

R2L saldırı etiketlerinin ¾ oranında normal örneklemesi sonucunda 1741 adet verinin 995 tanesi R2L saldırı, 746 tanesi normal etiketlidir. Buna göre model sonucu şu şekildedir:

	Accuracy	Precision	Recall	F1 Score
Voting Classifier	0.435	0.844	0.435	0.444

Örnekleme ile istenilen sonuç elde edilemediğinden bir sonraki yaklaşımda model ağırlıkları üzerine denemeler gerçekleştirilmiştir. Bu denemelerde ön işleme sonucu elde edilen saldırı veri setlerinden R2L veri seti ANNOVA F-Testinden elde edilen özellikler ile sınırlandırılarak olduğu gibi kullanılmıştır. Performans metriklerindeki *average* değerini ise *weighted* olarak kullanılmaya devam edilmiştir.

Bu denemede Logistic Regression ve Decision Tree Classifier modellerinin sınıf ağırlıkları *balanced* olarak ayarlanmıştır.

Ayrıca belirtilmelidir ki Adaboost ve Random Forest tasarımları neticesiyle dengesiz veri setlerinde tasarımları neticesiyle zaten iyi sonuç verdikleri bilinmektedir. Burada *n_estimators* parametresi ile zayıf sınıfa daha fazla odaklanılması sağlanılmıştır.

Geriye kalan Gaussian Naive Bayes ve Voting Classifier modellerine ise ek bir işlem yapılmamıştır. Aşağıda bunlara göre sonuçlar incelenmiştir,

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.723	0.665	0.723	0.685
Gaussian NB	0.850	0.851	0.850	0.828
Decision Tree Classifier	0.805	0.843	0.805	0.743
Adaboost	0.774	0.820	0.774	0.679
Random Forest Classifier	0.777	0.824	0.777	0.686
Voting Classifier	0.808	0.839	0.808	0.750

Bu adımın ardından istediğimiz sonuçlara ulaştığımızı gördük. Performans metriklerini *weighted* tutarak diğer seçili özellikler ile sınırlandırılmış veri setleri içinde aynı adımları uygulayarak sonuçları inceledik. Aşağıda tüm saldırı tiplerinin Voting Classifier sonuçları incelenmiştir:

	Accuracy	Precision	Recall	F1 Score
DoS	0.826	0.836	0.826	0.822
Probe	0.889	0.885	0.889	0.879
R2L	0.808	0.839	0.808	0.750
U2R	0.993	0.992	0.993	0.992

Bu son deneyimiz ile aradığımız sonuçlara ulaştığımızı gördük.

Son olarak performans metrikleri incelenmesi sonucu veri setlerindeki dengesizliğin büyük boyutlarda olduğu R2L ve U2R veri setleri için Recall, Probe ve DoS veri setleri içinse F1 Score değerlerinin dikkate alınmasının daha doğru olduğuna karar verilmiştir. Bu ayrıştırmadaki ana nedenimiz R2L ve U2R veri setlerindeki yüksek dengesizlik oranı nedeniyle zayıf saldırı sınıflarının gözden kaçmasını engellemektir. False Negative değerlerini minimize eden Recall metriğinin bu durum için uygun olduğuna karar verdik.

7. NİHAİ SONUÇLAR VE DEĞERLENDİRME

DOS için karşılaştırma tablosu:

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.816	0.824	0.816	0.811
Gaussian NB	0.813	0.814	0.813	0.811
Decision Tree Classifier	0.845	0.853	0.845	0.841
Adaboost	0.836	0.847	0.836	0.832
Random Forest Classifier	0.829	0.841	0.829	0.824
Voting Classifier	0.826	0.836	0.826	0.822

PROBE için karşılaştırma tablosu:

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.880	0.876	0.880	0.868
Gaussian NB	0.886	0.883	0.886	0.875
Decision Tree Classifier	0.884	0.881	0.884	0.873
Adaboost	0.892	0.889	0.892	0.883
Random Forest Classifier	0.888	0.885	0.888	0.878
Voting Classifier	0.889	0.885	0.889	0.879

R2L için karşılaştırma tablosu:

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.723	0.665	0.723	0.685
Gaussian NB	0.850	0.851	0.850	0.828
Decision Tree Classifier	0.805	0.843	0.805	0.743
Adaboost	0.774	0.820	0.774	0.679
Random Forest Classifier	0.777	0.824	0.777	0.686
Voting Classifier	0.808	0.839	0.808	0.750

U2R için karşılaştırma tablosu:

	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.960	0.991	0.960	0.974
Gaussian NB	0.992	0.992	0.992	0.992
Decision Tree Classifier	0.991	0.989	0.991	0.990
Adaboost	0.994	0.994	0.994	0.991
Random Forest Classifier	0.994	0.994	0.994	0.991
Voting Classifier	0.993	0.992	0.993	0.992

Tüm saldırıların Voting Classifier ile genel sonuçlarının tablosu aşağıda verilmiştir. Bu tabloda DoS ve Probe için F1 Score değeri R2L ve U2R saldırıları için Recall değeri göz önünde bulundurulmuştur.

	Voting Classifier
DoS	0.822
Probe	0.879
R2L	0.808
U2R	0.993

Değerli sonuçlar elde etmiş iki çalışmayı birleştirerek geliştirdiğimiz bu projede KDD veri setlerini işleyerek DoS, Probe, U2R ve R2L saldırıları için özel veri setleri elde ettik. Bu veri setleri ile ANNOVA F-Testi yaparak her veri setini ifade eden en iyi 13 özelliği çıkardık. Saldırı veri setlerini seçili özellikler ile sınırlandırarak çeşitli makine öğrenmesi modelleri eğittik. En son her saldırı için Voting Classifier Modeli ile kullandığımız makine öğrenmesi modellerini birleştirdik ve daha iyi sonuçlar almak için çalışmalar yaptık. Bu çalışmalarda model sonuçlarını iyileştirmek için birçok deney gerçekleştirdik. Model sınıf ağırlıkları, performans ortalama değerleri ve veri setini örnekleme üzerine birçok yaklaşım denedik. Bu çalışmalar sırasında veri setinin kalitesinin, modellerin parametrelerinin sonuçları nasıl etkilediğini inceledik.

Sonuç olarak başarılı performans metrikleri sergileyen modeller elde ettik. Projenin main.ipynb dosyasında son kısımda bulunan modeli test etmek için geliştirdiğimiz fonksiyon denendiğinde görülmektedir ki Voting Classifier modelleri çoğunlukla doğru sonuç vermekte, saldırı ve normal satırlarını doğru etiketleyebilmektedir.

Nihayetinde birçok çalışmayı inceleyerek önemli bilgiler edindiğimiz, iki değerli çalışmayı birleştirerek tatmin edici sonuçlar elde ettiğimiz başarılı bir projeyi tamamlamış olduk.

7.KAYNAKÇA

[1] - Kitsune IDS makalesi: https://arxiv.org/abs/1802.09089

 Kitsune IDS GitHub sayfası: https://github.com/topics/kitsune

[2] - Makalesi: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8494096

 GitHub sayfası: https://github.com/rahulvigneswaran/Intrusion-Detection-Systems?tab=readme-ov-file

[3] - Makalesi: https://thesai.org/Downloads/Volume7No4/Paper_19-A_Subset_Feature_Elimination_Mechanism_for_Intrusion_Detection_System.pdf

 GitHub sayfası: https://github.com/CynthiaKoopman/Network-Intrusion-Detection?tab=readme-ov-file

[4] - Makalesi: https://arxiv.org/pdf/2207.02117

 GitHub sayfası: https://github.com/othmbela/dbn-based-nids?tab=readme-ov-file

[5]- https://ilyasbinsalih.medium.com/what-is-hard-and-soft-voting-in-machine-learning-2652676b6a32

https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/