T.C.
MİLLÎ SAVUNMA ÜNİVERSİTESİ
ALPARSLAN SAVUNMA BİLİMLERİ VE MİLLÎ GÜVENLİK ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANS PROGRAMI

AĞ SALDIRI TESPİTİ İÇİN ÖZELLİK SEÇİMİ TEMELLİ MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ

YÜKSEK LİSANS TEZİ

EMRE EMİRMAHMUTOĞLU
2181806

TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ YILMAZ ATAY

ANKARA
HAZİRAN 2024

T.C.
MİLLÎ SAVUNMA ÜNİVERSİTESİ
ALPARSLAN SAVUNMA BİLİMLERİ VE MİLLÎ GÜVENLİK ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK YÜKSEK LİSANS PROGRAMI

AĞ SALDIRI TESPİTİ İÇİN ÖZELLİK SEÇİMİ TEMELLİ MAKİNE ÖĞRENMESİ ALGORİTMALARININ KARŞILAŞTIRMALI ANALİZİ

YÜKSEK LİSANS TEZİ

EMRE EMİRMAHMUTOĞLU
2181806

TEZ DANIŞMANI: DR. ÖĞR. ÜYESİ YILMAZ ATAY

ANKARA
HAZİRAN 2024

ÖZGÜNLÜK RAPORU

Tez çalışmamın a) Kapak sayfası, b) Giriş, c) Literatür Özeti, ç) Saldırı Tespit Sistemlerinin Sınıflandırılması, d) Materyal ve Metot, e) Deneysel Çalışmalar ve f) Sonuçlar ve Öneriler kısımlarından oluşan toplam 133 sayfalık kısmına ilişkin, 17/05/2024 tarihinde şahsım tarafından "Turnitin" adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı %10'dur.

Uygulanan filtrelemeler:

1- Kaynakça hariç

2- Alıntılar hariç

3- 5 kelimeden daha az örtüşme içeren metin kısımları hariç

Millî Savunma Üniversitesi Alparslan Savunma Bilimleri ve Millî Güvenlik Enstitüsü Lisansüstü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Usul ve Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Emre EMİRMAHMUTOĞLU

03/07/2024

ETİK BEYANI

Millî Savunma Üniversitesi Enstitüleri Lisansüstü Tez Hazırlama Kılavuzu'nda yer alan kurallarına uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bu tezdeki düşünce, görüş, varsayım, sav veya tezler bana aittir; Millî Savunma Bakanlığı, Millî Savunma Üniversitesi ve Alparslan Savunma Bilimleri ve Millî Güvenlik Enstitüsü sorumlu tutulamaz.

Emre EMİRMAHMUTOĞLU

03/07/2024

İmza

Ulu Önder Gazi Mustafa Kemal Atatürk'ün mezun olduğu okulda yüksek lisans eğitimi almanın gururuyla, desteğini hiçbir zaman esirgemeyen Sevgili Aileme…

ÖNSÖZ VE TEŞEKKÜR

Tez çalışmam kapsamında akademik bilgi birikimi, değerli yardımı, rehberliği ve katkılarıyla beni sürekli destekleyerek hedefe odaklanmamı sağlayan değerli hocam ve tez danışmanım Sayın Dr. Öğr. Üyesi Yılmaz ATAY'a, yüksek lisans eğitimim boyunca Bilgisayar Mühendisliği ve Siber Güvenlik alanında akademik bilgi ve tecrübeleri ile bana değer katan, saygıdeğer hocalarıma teşekkür ve saygılarımı sunarım.

Ayrıca, bu sürece başlamamda ve eğitim sürecim boyunca beni destekleyen ve cesaretlendiren Sayın Fatih Yel başta olmak üzere değerli ailem ve arkadaşlarıma, desteğini ve anlayışını bu süreç boyunca hissettiğim yöneticim ve iş arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Ankara; Mayıs 2024 Emre EMİRMAHMUTOĞLU

İÇİNDEKİLER

TABLOLAR LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

KISALTMALAR

AB **:** Adaboost, Adaptive Boosting (Uyarlanabilir Artırma)

ABC **:** Artificial Bee Colony (Yapay Arı Kolonisi)

ACCS **:** Australian Cyber Security Center (Avustralya Siber Güvenlik Merkezi)

ACK **:** Acknowledgement (Onay)

ADASYN **:** Adaptive Synthetic (Uyarlanabilir Sentetik)

AFS **:** Artificial Fish Swarm (Yapay Balık Sürüsü)

AE **:** Auto Encoder (Otomatik Kodlayıcı)

AGEO **:** Adaptive Grasshopper Optimization Algorithm (Uyarlanabilir Çekirge Optimizasyon Algoritması)

ALO **:** Ant Lion Optimization (Karınca Aslanı Optimizasyon)

ANN **:** Artificial Neural Network (Yapay Sinir Ağları)

AP **:** Affinity Propagation (Benzeşim Yayılımı)

BG **:** Bagging (Torbalama)

BN **:** Bayes Network (Bayes Ağı)

BP **:** Back Propagation (Geri Yayılım)

C4.5 **:** C4.5 Decision Tree (C4.5 Karar Ağacı)

CART **:** Classification And Regression Tree (Sınıflandırma Ve Regresyon Ağacı)

CAIDA **:** Center for Applied Internet Data Analysis (Uygulamalı İnternet Veri Analizi Merkezi)

CFS **:** Correlation Based Feature Selection (Korelasyona Dayalı Özellik Seçimi)

CFS-BA : Correlation Based Feature Selection Bat Algorithm (Korelasyon Tabanlı Özellik Seçimi Yarasa Algoritması)

CIC **:** Canadian Institute of Cybersecurity (Kanada Siber Güvenlik Enstitüsü)

CNN **:** Convolutional Neural Network (Evrişimli Sinir Ağları)

DBN **:** Deep Believe Network (Derin İnanç Ağları)

DDoS **:** Distributed Denial of Service (Dağıtık Hizmet Reddi)

DE **:** Differantial Evolution (Diferansiyel Evrim)

DEA **:** Differential Evolution Algorithm (Diferansiyel Evrim Algoritması)

DL : Deep Learning (Derin Öğrenme)

DNN : Deep Neural Network (Derin Sinir Ağları)

DT : Decision Tree (Karar Ağacı)

DoS : Denial of Service (Hizmet Reddi)

EA : Evolution Algorithm (Evrimsel Algoritma)

EFS : Ensemble of Feature Selection (Özellik Seçimi Topluluğu)

FAR : False Alarm Rate (Yanlış Alarm Oranı)

FCM : Fuzzy C-Means Clustering (Bulanık C-Ortalama Kümeleme)

FIN : Finish (Bitiş)

FN : False Negative (Yanlış Negatif)

FNR : False Negative Rate (Yanlış Negatif Oranı)

FP : False Positive (Yanlış Pozitif)

FPA : Flower Pollination Algorithm (Çiçek Tozlaşma Algoritması)

FPR : False Positive Rate (Yanlış Pozitif Oranına)

FTP : File Transfer Protocol (Dosya Aktarım Protokolü)

GA : Genetic Algorithm (Genetik Algoritma)

GANBADM : Genetic Algorithm Wrapper-Based feature selection and Nave Bayes for Anomaly Detection Model (Genetik Algoritma Sarmalayıcı Tabanlı özellik seçimi ve Anomali Tespit Modeli için Saf Bayes)

gbest : Global Best Solution (Global En İyi Çözüm, Uygunluk Fonksiyonunun En İyi Değeri)

GbFS : GA Based Feature Selection (GA Tabanlı Özellik Seçimi)

GB : Gradient Boosting (Gradyan Artırma)

GB-EGWO : Genetic Based Enhanced Grey Wolf Optimization (Genetik Tabanlı Geliştirilmiş Gri Kurt Optimizasyonu)

GBT : Gradient Boosting Tree (Gradyan Güçlendirme Ağacı)

GIWRF : Gini Impurity-Based Weighted Random Forest (Gini Safsızlık Tabanlı Ağırlıklı Rastgele Orman)

GMM : Gaussian Mixture Model (Gauss Karışım Modeli)

GRU : Gated Recurrent Unit (Geçitli Tekrarlayan Birim)

GSA : Gravitational Search Algorithm (Yerçekimsel Arama Algoritması)

HKELM : Hybrid Kernel Function Extreme Learning Machine (Hibrit Çekirdek Fonksiyonuna Sahip Bir Aşırı Öğrenme Makinesi)

HMLSTM : Hierarchical Multiscale LSTM (Hiyerarşik Çok Ölçekli LSTM

HTTP	**:** Hyper Text Transfer Protokol (Hiper Metin Transfer Protokolü)
HTTPS	**:** Güvenli Hiper Metin Transfer Protokolü (Secure Hyper Text Transfer Protokol)
ICMP	**:** Internet Control Message Protocol (İnternet Kontrol Mesajı Protokolü)
IDS	**:** Intrusion Detection System (Saldırı Tespit Sistemi)
IE	**:** Information Entropy (Bilgi Entropisi)
IE-DBN	**:** Information Entropy Deep Believe Network (Bilgi Entropisi Derin İnanç Ağları)
IG	**:** Information Gain (Bilgi Kazancı)
IGR	**:** Information Gain Ratio (Bilgi Kazanç Oranı)
IMAP	**:** Internet Message Access Protocol (İnternet Mesaj Erişim Protokolü)
IoT	**:** Internet of Things (Nesnelerin İnterneti)
I-OVO	**:** Improved One-vs-One (Geliştirilmiş Bire Karşı Bir)
IP	**:** Internet Protocol (İnternet Protokolü)
IRC	**:** İnternet Aktarmalı Sohbet Protokolü (Internet Relay Chat Protocol)
ISCX	**:** Information Security Center of Excellence (Bilgi Güvenliği Mükemmeliyet Merkezi)
J48	**:** J48 Decision Tree (J48 Karar Ağacı)
KNN	**:** K Nearest Neighbours (K En Yakın Komşu)
KPCA	**:** Kernel Principal Component Analysis (Çekirdek Temel Bileşen Analizi Algoritması)
KSIDO-PABCE	**:** Kulczynski Similarity Indexed Dragonfly Optimization-Based Polytomous Adaptive Base Class Ensemble (Kulczynski Benzerlik İndeksli Yusufçuk Optimizasyonuna Dayalı Çok Kategorili Uyarlanabilir Temel Sınıf Topluluğu)
LBNL	**:** Lawrence Berkeley National Laboratory (Lawrence Berkeley Ulusal Laboratuvarı Veri Kümesi)
LDA	**:** Linear Discriminant Analysis (Doğrusal Diskriminant Analizi)
LightGBM	**:** Light Gradient-Boosting Machine (Hafif Gradyan Arttırma Makinesi)
LNNLS-KH	**:** Linear Nearest Neighbor Lasso Step Optimization (Kril Sürüsü Doğrusal En Yakın Komşu Lasso Adım Optimizasyonuna)
LR	**:** Logistic Regression (Lojistik Regresyon)
LSO	**:** Lion Swarm Optimization (Aslan Sürüsü Optimizasyonu)

LSTM : Long Short-Term Memory (Uzun Kısa Süreli Bellek)

LSTM-RNN : Long Short-Term Memory Recurrent Neural Network (Uzun Kısa Süreli Bellek Tekrarlayan Sinir Ağları)

MABA : Modified Adaptive Boosting with Area Under The Curve Algorithm (Eğri Algoritması Altındaki Alanla Değiştirilmiş Uyarlanabilir Güçlendirme)

MIFA : Mutual Information Firefly Algorithm (Karşılıklı Bilgi Ateşböceği Algoritması)

MIT : Massachusetts Institute of Technology (Massachusetts Teknoloji Enstitüsü)

ML : Machine Learning (Makine Öğrenmesi)

MLP : Multi Layer Perceptron (Çok Katmanlı Algılayıcı)

MLP-NN : Multi Layer Perceptron Neural Network (Çok Katmanlı Algılayıcı Sinir Ağı)

MOGA : Multi-Objective Genetic Algorithm (Çok Amaçlı Genetik Algoritma)

MSCNN-LSTM : Multiscale Convolutional Neural Network Long Short-Term Memory (Çok Ölçekli Evrişimli Sinir Ağı Uzun Kısa Süreli Bellek)

NB : Naive Bayes (Saf Bayes)

NMAP : Network Mapper (Ağ Haritalayıcı)

NN : Neural Network (Sinir Ağı)

OCNN : Optimized CNN (Optimize Edilmiş CNN)

OCSVM : One Class Support Vector Machine (Tek Sınıf Destek Vektör Makinesi)

OpenSSH : Açık Güvenli Kabuk Protokolü (Open Secure Shell Protocol)

pbest : Personal Best Solution (Kişisel En İyi Çözüm)

PCA-GWO : Principal Component Analysis with Grey Wolf Optimization (Gri Kurt Optimizasyonu ile Temel Bileşen Analizi)

PCAP : Packet Capture (Paket Yakalama)

RC-NN : Recurrent Convolutional Neural Network (Tekrarlayan Konveksiyonel Sinir Ağı)

REPTree : Reduced-Error Pruning Tree (Azaltılmış Hata Budama Ağacı)

PERL : Practical Extraction and Report Language (Pratik Çıkarım ve Raporlama Dili)

PIO : Pigeon Inspired Optimizer (Güvercinden İlham Alan Bir Optimize Edici)

POP3 : Post Office Protocol 3 (Postane Protokolü 3)

PSO	**:** Particle Swarm Optimization (Parçacık Sürüsü Optimizasyonu)
QDA	**:** Quadratic Discriminant Analysis (İkinci Dereceden Diskriminant Analizi)
R2L	**:** Remote to Local (Uzaktan Yerele)
RBM	**:** Restricted Boltzmann Machine (Kısıtlı Boltzmann Makinesidir)
ResNet	**:** Residual Network (Artık Ağ)
RF	**:** Random Forest (Rastgele Orman)
RFE	**:** Recursive Feature Elimination (Özyinelemeli Özellik Eleme)
RMSE	**:** Root Mean Square Error (Karekök Ortalama Hatasını)
RNN	**:** Recurrent Neural Network (Tekrarlayan Sinir Ağları)
SAE	**:** Sparse Autoencoder (Seyrek Otomatik Kodlayıcı)
SDAE-ELM	**:** Stacked Denoising Autoencoder Extreme Learning Machine (Yığılmış Gürültü Giderici Otomatik Kodlayıcı Aşırı Öğrenme Makinesi)
SMOTE	**:** Syntetic Minority Over Sampling Technique (Sentetik Azınlık Aşırı Örnekleme Tekniği)
SMTP	**:** Simple Mail Transfer Protocol (Basit Posta Aktarım Protokolü)
SS	**:** Özellik Seçimi Uygulanmadan Veri Setlerinin Makine Öğrenmesi Algoritmaları ile Sınıflandırılması
SSH	**:** Secure Shell Protocol (Güvenli Kabuk Protokolü)
STS	**:** Saldırı Tespit Sistemi
SVM	**:** Support Vektor Machine (Destek Vektör Makinesi)
SYN	**:** Synchronize (Senkronize)
TCP	**:** Transmission Control Protocol (Geçiş Kontrol Protokolü)
TN	**:** True Negative (Gerçek Negatif)
TP	**:** True Positive (Gerçek Pozitif)
TPR	**:** True Positive Rate (Doğru Pozitif Oranı)
TS	**:** Tabu Search (Tabu Arama)
U2R	**:** User to Root (Kullanıcıdan Kök Kullanıcıya)
UDP	**:** User Datagram Protocol (Kullanıcı Veribloğu İletişim Kuralları)
UNB	**:** New Brunswick University (New Brunswick Üniversitesi)
VS1	**:** KDD CUP 99 Veri Seti
VS2	**:** NSL KDD Veri Seti
VS3	**:** UNSW-NB15 Veri Seti

VS4 **:** CSE-CIC-IDS2018 Veri Seti

XGBoost **:** Extra Gradient Boost (Ekstra Gradyan Artırma)

YZ **:** Yapay Zeka

ÖZ

Ağ Saldırı Tespiti için Özellik Seçimi temelli Makine Öğrenmesi Algoritmalarının Karşılaştırmalı Analizi

Emre EMİRMAHMUTOĞLU

Millî Savunma Üniversitesi, Alparslan Savunma Bilimleri ve Millî Güvenlik Enstitüsü

Ankara, Haziran 2024

Teknolojinin sürekli gelişimiyle birlikte küresel çapta internet kullanımı önemli ölçüde artmıştır. Bu artış küresel ağlarda dolaşan hassas, kişisel ve ticari veri miktarını doğrusal olarak artırmaktadır. Kötü niyetli girişimler pek çok ağ saldırı türleri ortaya çıkardığından, siber güvenlik bağlamında ağ ve bilgi işlem cihazlarının güvenliğini sağlama gereksinimi ortaya çıkmaktadır. Bu gereksinim; ağı izleyen ve ağ trafiği aracılığıyla şüpheli ve kötü amaçlı etkinlikleri veya politika ihlallerini belirleyen, bu da ağ yöneticilerinin mevcut tehditleri sürekli izlemesine olanak tanıyan Saldırı Tespit Sistemi-STS'lerini güvenlik mimarisinin önemli bir bileşeni haline getirmiştir. STS, ortama, yerleşime ve veri kaynağına bağlı olarak ana bilgisayar tabanlı ve ağ tabanlı olarak sınıflandırılır. Ana bilgisayar tabanlı STS, ana bilgisayara STS kurulma dezavantajı oluşturduğundan; kullanım ve performans açısından ağ tabanlı STS daha çok tercih edilmektedir. STS'ler saldırı tespit yaklaşımı bakımından anormallik ve imza tabanlı olarak sınıflandırılmaktadır. İmza tabanlı STS mevcut kütüphanesinde bulunan örnekler üzerinden analiz ederden, anormallik tabanlı STS makine öğrenmesi yöntemleri ile analiz gerçekleştirir. Bu tezde PSO, FPA, DE özellik seçimi yöntemleri ile LR, DT, RF, KNN, NB, GB, LDA, QDA, AdaBoost, NN makine öğrenmesi algoritmaları kullanılarak KDD Cup 99, NSL-KDD, UNSW-NB15, CSE-CIS-IDS2018 veri setleri üzerinde anormallik tabanlı karşılaştırmalı analiz çalışması ortaya konulmuştur.

Anahtar Sözcükler : Siber Güvenlik, Saldırı Tespit Sistemi, Makine Öğrenmesi, Özellik Seçimi, Bilgisayar Ağları, Sınıflandırma.

Bilim Kodu : 92438

Sayfa Sayısı : 133

Tez Danışmanı : Dr. Öğr. Üyesi Yılmaz ATAY

ABSTRACT

Comparative Analysis of Machine Learning Algorithms based on Feature Selection for Network Intrusion Detection

Emre EMİRMAHMUTOĞLU

National Defence University, Alparslan Defence Sciences and National Security Institue

Ankara, June 2024

With the continuous development of technology, global internet usage has increased significantly. This increase linearly increases the amount of sensitive, personal and commercial data travelling on global networks. As malicious attempts create many types of network attacks, the need to ensure the security of network and computing devices in the context of cyber security arises. This requirement has made Intrusion Detection Systems (IDS), which monitor the network and identify suspicious and malicious activities or policy violations through network traffic, allowing network administrators to continuously monitor current threats, an important component of the security architecture. IDS are classified as host-based and network-based depending on the environment, placement and data source. Since host-based IDS has the disadvantage of installing IDS on the host computer, network-based IDS is more preferred in terms of usage and performance. IDSs are classified as anomaly and signature based in terms of intrusion detection approach. While signature-based IDS analyses the samples in the existing library, anomaly-based IDS performs analysis with machine learning methods. In this thesis, PSO, FPA, DE feature selection methods and LR, DT, RF, KNN, NB, GB, LDA, QDA, AdaBoost, NN machine learning algorithms are used for anomaly-based comparative analysis on KDD Cup 99, NSL-KDD, UNSW-NB15, CSE-CIS-IDS2018 datasets.

Keywords : Cyber Security, Intrusion Detection System, Machine Learning, Feature Selection, Computer Networks, Classification.

Science Code : 92438

Pages : 133

Supervisor : Asst. Prof. Yılmaz ATAY

1. GİRİŞ

Küresel ağda dolaşan önemli, hassas ve gizli kişisel ve ticari verilerin miktarı, internet kullanımındaki artışın etkisiyle önemli ölçüde artmıştır. Ağ saldırı türlerinin güncel olarak evrimi ve bilgi işlem cihazları arasındaki veri alışverişindeki artış, ağ ve bilgi işlem cihazlarının güvenliğini sağlama gereksinimini ortaya çıkarmaktadır. Küresel IP trafiğinin 2022 yılına kadar üç kat artacağı ve bunun da yıllık büyümede %26 artışa tekabül edeceği düşünülmektedir (Cisco Mobile Visual Networking Index (VNI), 2017). Kişi başına toplanan net iş trafiği, 2017 yılında kaydedilen aylık 16 GB olup, 2022 yılına kadar 50 GB'a yükselmesi beklenmektedir (Cisco Mobile Visual Networking Index (VNI), 2017). Günümüzde, öngörüldüğü gibi bu trafik daha da yüksek seviyelere gelmiştir. Ağda bilgi alışverişi yapmak, veri toplamak veya ilişkili faaliyetleri izlemek için kullanılan araçlar saldırı yüzeyini arttırır. Saldırganlar, sistemlerin çalışmasını etkileyebilecek, verilerin kullanılabilirliğini ve gizliliğini değiştirebilecek hassas verilere erişim sağlamak için güvenlik önlemlerindeki kusurlar yoluyla ağın güvenliğini ihlal etme girişimlerinde bulunurlar. Dolayısıyla siber güvenlik, siber saldırılarla mücadelede ve buna bağlı maliyet ve zararların azaltılmasında kritik önlemler sunan önemli bir alan olarak görülmeye başlanmıştır. Bu amaçla güvenlik mimarisinin hayati bir bileşeni olan Saldırı Tespit Sistemi (IDS), çeşitli izinsiz giriş türlerini tanıyan bir araç görevi görmektedir. STS bir ağı izler ve ağ trafiği aracılığıyla şüpheli ve kötü amaçlı etkinlikleri veya politika ihlallerini belirler; bu da ağ yöneticilerinin mevcut tehditleri sürekli izlemesine olanak tanımaktadır. Buradan yola çıkılarak, ağı saldırılara karşı koruyabilecek verimli bir STS sistemi kurma gereksinimi ortaya çıkmıştır.

Yıllar içinde STS'ler, ağ trafiğinin meşru olduğundan ve kötü amaçlı olmadığından emin olmak için geliştirilmiştir. Yeni nesil güvenlik duvarı, ağ saldırılarının karmaşıklığını gidermek için geliştirilmiştir. İmzalara, davranış analizine ve kötü niyetli faaliyetlere dayalı olarak izinsiz girişleri tespit etmek için STS modülünü entegre etmiştir. STS terimi, ağ ortamındaki herhangi bir anormal davranışta alarm üreten sistem olarak adlandırılabilir. STS'nin birincil görevi, sistem güvenliğini

tehlikeye atan davetsiz misafirleri veya kötü niyetli etkinlikleri önlemek için çeşitli işlem kombinasyonları uygulayarak bir sistemi korumaktır.

STS, ortama, yerleşime ve veri kaynağına bağlı olarak ana bilgisayar tabanlı ve ağ tabanlı olarak sınıflandırılır. Ana bilgisayar tabanlı STS, yalnızca şüpheli eylemleri ve güvenlik politikası ihlallerini tarayarak cihazdaki veri paketlerini izlemek için tek bir ana bilgisayara veya cihaza dağıtılır. Bu yaklaşımın dezavantajı, korunması gereken her ana bilgisayara bir STS kurma zorunluluğudur. Ayrıca, ana bilgisayar tabanlı bir STS muhtemelen her ana bilgisayar için daha fazla işlem süresi gerektirecek ve bu da sonuçta performansını düşürecektir. Buna karşılık, ağ tabanlı STS, kötü amaçlı etkinlikleri tespit etmek amacıyla trafiği izlemek, yakalamak ve analiz etmek için bilgisayar ağının tamamına dağıtılır. STS'ler saldırı tespiti için anormallik tabanlı veya kötüye kullanıma dayalı (imza tabanlı) olmak üzere iki temel analiz yaklaşımı kullanır. Bir STS, bilinen saldırıların imzalarını veya önceden tanımlanmış normal aktivite profilinden sapmaları arayarak çalışır. Tipik olarak, "kötüye kullanım izinsiz giriş tespiti" veya "imza tabanlı izinsiz giriş tespiti" olarak da adlandırılan STS, reaktiftir ve saldırıları daha sonra bir saldırı veri tabanı olarak tutulan imzalar ve modeller biçiminde sunar. Bu imzalar/modeller daha sonra ağ üzerinden alınan verilerle karşılaştırılır; bir eşleşme varsa sistem bu tür etkinlikleri kötü amaçlı olarak işaretler. Bu yaklaşımın geliştirilmesi kolaydır ve bilinen saldırıları çok düşük yanlış pozitiflerle kolayca tanımlama avantajına sahiptir. Ancak yeni/sıfır gün saldırı türlerini tespit etme yeteneğinden yoksundur. Bunun nedeni yalnızca veri tabanındaki saldırıları tespit edebilmesidir. Ayrıca olası saldırılara karşı sürdürülmesi, güncellenmesi ve veri paketleriyle karşılaştırılması gereken geniş imza veri tabanı nedeniyle çok büyük kaynak tüketir. Aksine, genellikle "davranış tabanlı STS" olarak bilinen anormallik tabanlı saldırı tespit sistemleri, sistemin normal davranışının bir modelini oluşturur, Ardından modeldeki düzensizliklere dayalı saldırıları tanımlamak için modelden sapan etkinlikleri arar. Bu yaklaşımın gücü, yeni bilinmeyen saldırıları tespit etme yeteneğinde yatmaktadır; ancak genellikle daha yüksek bir yanlış pozitif oranına yol açar ve bu da uygulamasını sınırlar. Ağdaki ve bilgisayardaki tehditlerin listesi sonsuzdur ve sürekli olarak gelişmektedir; bu nedenle, izinsiz giriş tespiti aktif bir araştırma alanı olmaya devam etmektedir.

Araştırmacılar, saldırıları etkili bir şekilde tespit edebilecek STS yöntemleri geliştirmek için birçok girişimde bulunmuşlardır. Çok azı kötüye kullanım veya imza

tabanlı yaklaşımı temel alırken, yakın zamanda literatürde sunulan çoğu STS, izinsiz girişleri tanımlamak için makine öğrenmesi tekniklerini yani yapay zekayı kullanan anormallik tabanlı yaklaşıma odaklanmıştır.

Bu çalışmada makine öğrenmesi algoritmaları ile veri setlerinde bulunan saldırıların tespitine odaklanılmıştır. Literatürde yoğun olarak üzerinde çalışılan KDD CUP 99, NSL-KDD, UNSW-NB15 ve CICIDS2018 veri setleri kullanılmıştır. Meta sezgisel algoritma olan; parçacık sürü optimizasyonu, çiçek tozlaşma ve diferansiyel evrim algoritmaları ile özellik seçimi yaklaşımı benimsenmiştir. Bu veri setleri üzerinden, makine öğrenmesi algoritmaları ile özellik seçimi yapılarak ve özellik seçimi yapılmadan sınıflandırma sonuçları karşılaştırmalı olarak ortaya konmuştur. Veri setleri üzerinden makine öğrenmesi algoritmalarının karşılaştırmalı bir analizinin ortaya konulması amaçlanmıştır.

2. LİTERATÜR ÖZETİ

2.1. Ağa İzinsiz Giriş Kategorileri

Ağa izinsiz giriş kategorilerinden bazıları şunlardır(Rahul et al., 2018):

- **Hizmet Reddi (DoS):** Bu tür izinsiz girişte, saldırgan, yüksek hacimli istekler göndererek ana bilgisayara aşırı yükleme yapar. Bu eylem, ana bilgisayarın hizmetlerinin geçici veya kalıcı olarak kesintiye uğramasına yol açar.
- **Kullanıcıdan Kök Kullanıcıya (U2R):** Bu tür izinsiz girişlerde saldırgan, root (kök kullanıcı)'u yönetmek için ağ deliklerini kullanmayı amaçlar. Bu eylem, bir kullanıcıyı hedefleyerek ve onun ağa erişimini sağlayarak gerçekleştirilir.
- **Uzaktan Yerele (R2L):** Bu, saldırganın mevcut kullanıcı olarak yerel erişim elde etmeyi hedeflediği bir izinsiz giriş türüdür. Bunu yapabilmek için saldırgan, aslında gerçek bir hesap yokken makineye veri paketleri göndererek bir güvenlik açığını arar ve kullanır.
- **Derinlemesine Araştırma (Probe):** Bu tür izinsiz girişte saldırgan, ağdaki bilgisayarlar gibi belirli bilgileri toplayarak root erişimi elde etmeye çalışır.

2.1.1. Hizmet Reddi

Hizmet Reddi (DoS) kategorisi çeşitli saldırılardan oluşur. Bunlardan bazıları aşağıdaki gibidir:

Smurf: Smurf saldırısı, sahte yayın ping mesajları yoluyla hedef sistemde kayda değer ağ trafiği oluşturur. Bu saldırı türünde saldırgan, hedefin sahte kaynak IP adresi de dahil olmak üzere ağdaki farklı adreslere yankı istekleri gönderir. Ana bilgisayarlar isteğe yanıt verdikçe hedef ağ büyük bir trafikle dolacaktır (Zargar & Kabiri, 2009).

Neptune: Bu saldırıda saldırgan, hedefin kendisine çok sayıda TCP bağlantısı göndererek diğer ana bilgisayarlardan TCP bağlantısı almasını imkansız hale getirir (I.Ahmad et al., 2009).

Land: Bu saldırıda saldırgan hedefe sahte bir SYN paketi gönderir. Bu, sık sık kendisiyle senkronize olmasına neden olur (I. Ahmad et al., 2009).

Back: Back saldırı, saldırganın isteklerinin birkaç ön eğik çizgi içerdiği Apache web sunucularına karşı yapılan bir saldırıdır. Bu isteklerin işlenmesi çok zaman gerektirir (I. Ahmad et al., 2009).

Ping of Death: Bu saldırıda saldırgan, hedef ana bilgisayara ping istekleri olarak büyük boyutlu ICMP paketleri gönderir. Bu, hedef ana bilgisayarın kullanılamamasına neden olacaktır (I. Ahmad et al., 2009).

Teardrop: Bu saldırıda saldırgan, eski işletim sistemleri tarafından yanlış yönetilen, çakışan IP parçaları gönderir. Bu, hedef konağın kafa karışıklığına yol açacaktır (I. Ahmad et al., 2009).

2.1.2. Kullanıcıdan Kök Kullanıcıya

U2R saldırılarından bazıları şunlardır:

Arabellek Taşması: Bu saldırı, verilerin uygunluğuna yönelik herhangi bir inceleme yapılmadan çok sayıda verinin statik bir arabelleğe kopyalanmasıyla gerçekleşir (Marinova-Boncheva, 2007).

Yük Modülü (Load Module): SunOS 4.1 sistemlerine, loadmodule programından yararlanan bir saldırıdır. Sonuç olarak saldırgan root erişimi elde edebilir (Akasapu, 2017).

PERL: Bu, kök erişimi elde etmek için Perl programlama dili yürütmelerindeki bir kusurdan yararlanan bir saldırıdır (Akasapu, 2017).

Kök Kullanıcı Takımı (Rootkit): Bu saldırıda saldırgan, belirli bir sistemdeki bir güvenlik açığından yararlanarak veya bir parolayı kırarak kullanıcı düzeyinde erişim elde eder ve ardından saldırganın bir bilgisayara kontrollü erişim elde etmesine olanak tanıyan bir grup yazılım aracı olan Rootkit'i yükler (Akasapu, 2017).

2.1.3. Uzaktan Yerele

R2L kategorisi birçok farklı saldırıyı içerir. Bunlardan bazıları aşağıdaki gibidir:

FTP Yazma: Bu saldırıda saldırgan, yerel erişim elde edebilmek için bazı dosyaları ftp kök dizinine ekler (M. Xiao & Xiao, 2007).

Şifreyi Tahmin Etme: Bu saldırıda saldırgan, ağa yerel erişim sağlamak için misafir kullanıcının şifresini bulur. Parolanın tahmin edilmesi genellikle kolaydır veya hiç yoktur (Akasapu, 2017).

IMAP: Bu saldırıda saldırgan, yerel erişim elde etmek için güvenlik açığını kötüye kullanmak amacıyla bir makineye paketler iletir (Wutyi & Thwin, 2016).

PHF: Bu saldırı, bir sistemin bazı kaynaklarına erişmeye çalışır. Örneğin, bir web sunucusu, yerel yönetici olarak, kişinin yetkili olduğundan emin olmadan birine komut kabuğu erişimi veriyorsa, bu bir PHF betiği saldırısı olacaktır. Bu saldırı yerel bir kullanıcı veya uzak bir kullanıcı tarafından gerçekleştirilebilir (Wutyi & Thwin, 2016).

Warezmaster: Warezmaster, FTP sunucusunun yanlış erişim izni ile yapılandırılmasından kaynaklanan bir hata kullanır. Yani, FTP sunucusuna erişim yetkisi olan kişilere okuma yetkisi yerine yazma yetkisi verildiğinde ortaya çıkar. Kullanıcılar genellikle bu izne sahip değildir; bu, sunucuya dosya yükleme izninin olmadığı anlamına gelir. Bu saldırıda saldırgan, sunucuya bağlanmak için bir misafir hesabı kullanır. Saldırgan gizli bir dizin oluşturarak warez adı verilen yasa dışı yazılımların kopyalarını yükleyebilecektir. Sonuç olarak, diğer kullanıcılar bu dosyaları indirecektir. Kullanıcılara doğru izinleri vermek bu saldırıyı önlemenin kolay bir yoludur (Sabhnani & Serpen, 2003).

Warezclient: Warezmaster saldırısı tamamlandıktan sonra herhangi bir kullanıcı Warezclient saldırısı gerçekleştirme olanağına sahip olacaktır. Bu saldırıda kullanıcılar daha önce Warezmaster saldırısı sırasında yüklenen warez yazılımını indirebilir. Bu saldırı, dosyalar FTP sunucusundan indirileceği için yasal bir süreçmiş gibi görünmeye çalışmaktadır. Ancak bu saldırıyı ortaya çıkarabilecek bir gerçek vardır. Bu saldırı, kullanıcıların düzenli olarak konuk kullanıcılar tarafından kullanılamayan dizinlerden dosya indirdiğinin farkına varılarak tespit edilebilir (Sabhnani & Serpen, 2003).

2.1.4. Derinlemesine Araştırma

Probe saldırılarından bazıları şunlardır (Akasapu, 2017):

Ipsweep: Bu saldırıda saldırgan, güvenlik açıklarını tespit etmek için ağdaki ana bilgisayarları keşfeder.

Nmap: Nmap, FIN, SYN, ACK vb. gibi çeşitli tarama yöntemlerini kullanarak ağdaki açık ve kapalı portları keşfeden bir araçtır.

Portsweep: Bu saldırıda saldırgan ağdaki bir makinenin açık portlarını keşfeder.

Satan (Şeytan): Ağları Analiz Etmek için Güvenlik Yöneticisi Aracı (Şeytan), bir sistemdeki boşlukları araştıran bir araçtır.

2.2. Literatür Araştırması

Literatür araştırması kapsamında saldırı tespit sistemleri üzerine genel bir araştırma yapılmış, farklı veri tabanlarından saldırı tespit sistemlerinin ortaya çıkışı, çalışma prensibi, yapısına göre kullanım alanları, saldırı kategorileri ve saldırı çeşitleri hakkında bilgi edinilmiştir.

Mevcut çalışmalardan yola çıkılarak, saldırı tespit sistemlerinin siber güvenliğe etkisi üzerine odaklanılmıştır. Araştırma kapsamı ağ ve anormallik tabanlı saldırı tespit sistemleri olarak daraltılmıştır. Bu kapsamda; akademik ve açık kaynaklara başvurularak, literatürde var olan veri setleri ve bunlar üzerinde yapılan makine öğrenmesi algoritmaları ile saldırı tespiti çalışmalarına odaklanılmıştır.

Devan ve Khare (2010), özellik seçimi için Ekstra Gradyan Artırma (XGBoost) tekniğinin kullanıldığı ve NSL-KDD veri kümesinin ağ izinsiz girişlerini tespit etmek için DNN' i eğitmek için kullanıldığı STS için bir XGBoost-DNN modelini uygulamıştır (Devan & Khare, 2020). Modelin sonuçları mevcut Saf Bayes, lojistik regresyon ve destek vektör makineleri ile karşılaştırılmıştır. Sonuç, DNN'nin %97'lik bir skorla sınıflandırma doğruluğu açısından mevcut modellerden daha iyi performans gösterdiğini ortaya koymuştur.

Hajisalem ve Babaie (2018), benzersiz bir hibrit sınıflandırma için Yapay Arı Kolonisi (ABC) ve Yapay Balık Sürüsü (AFS) algoritmalarını birleştirmiştir (Hajisalem & Babaie, 2018). NSL-KDD ve UNSW-NB15 veri kümelerinde model değerlendirilmiştir. İlgisiz özellikleri çıkarırken eğitim veri setini ayırmak için Korelasyona Dayalı Özellik Seçimi (CFS) ve Bulanık C-Ortalama Kümeleme (FCM) teknikleri kullanılmıştır. Ayrıca, seçilen özelliklere dayalı olarak; modeli eğitmek, anormal ve normal kayıtları ayırt etmek adına kullanılan bir eğer-o halde (if-then) kuralları oluşturmak için Sınıflandırma ve Regresyon Ağacı (CART) tekniği kullanılmıştır. Elde edilen sonuçlar, önerilen yöntemin %99,9 doğruluk oranına, %0,01 yanlış alarm oranına ve %99,99 tespit oranına sahip olduğunu göstermiştir. Önerilen yöntemin ek yükü, hesaplama karmaşıklığı ve zaman maliyetlerinin incelenmesine göre mevcut ilgili metodolojiyle karşılaştırılabilir düzeydedir.

Vijayanand ve diğerleri (2018) tarafından, STS sınıflandırıcısının performansı, tüm saldırılarda ortak olan özellikler yerine her bir saldırı kategorisi için ilgili özelliklerin seçilmesiyle geliştirilmiştir (Vijayanand et al., 2018). Bu model genetik algoritmaya (GA) ve çoklu SVM sınıflandırıcılarına dayanmaktadır. CICIDS2017 ve ADFA-LD veri kümeleri temel alınarak önerilen sistemin performansı değerlendirilmiştir. %99 doğruluk oranına sahip olan model, daha az bilgi işlem karmaşıklığı ve iletişim ek yükü gerektirirken saldırıları doğru bir şekilde tanımlamak için STS için GA tabanlı özellik seçiminin kullanılması olasılığını göstermiştir.

Selvakumar B & Muneeswaran K (2018) tarafından önerilen STS modelinde özellik seçimi için sarmalayıcıda, Karşılıklı Bilgi Ateşböceği Algoritması (MIFA) kullanılmıştır (Selvakumar B & Muneeswaran K, 2018). Sınıflandırıcı olarak ise Bayes Ağı (BN) ile C4.5 Ağacı (C4.5) kullanılmış ve KDD CUP 99 veri setinde model değerlendirilmiştir. KDD CUP 99'a dayalı deneyler, önerilen çalışmanın hem doğruluğu hem de yanlış alarm oranlarını iyileştirdiğini ortaya çıkarmıştır. Model, C4.5 için sırasıyla %99,98, %63,85, %98,73, %17,24 ve BN için %99,95, %93,42, %97,83, %68,97 doğruluk elde etmiştir. Ayrıca DoS için 0,01, Probe için 0,04, R2L için 0,04 ve U2R saldırıları için 0,04'lük yanlış pozitif oranına (FPR) da ulaşmıştır. Ancak önerilen paradigma uzun hesaplamalar gerektirmektedir.

Mazini ve diğerleri (2018), yapay arı kolonisi ve AdaBoost algoritmalarına dayanan hibrit bir strateji önermiştir (Mazini et al., 2018). En uygun özellikleri seçmek için ABC kullanılmış ve seçilen özellikleri kategorize etmek için AdaBoost algoritması kullanılmıştır. Yapılan çalışmada, önerilen teknik NSL-KDD ve ISCXIDS2012 veri setleri üzerinde test edildiğinde %98,9 doğruluk elde edilmiştir. Önerilen teknik, bu veri setleri üzerinde yapılan diğer çalışmalar ile karşılaştırıldığında daha iyi performans göstermiştir. Ancak kullanılan veri seti günümüz ağlarına yönelik saldırılarını yansıtmamaktadır.

Aliawarneh ve diğerleri (2018), STS' deki yüksek yanlış negatif ve yanlış pozitif zorlukları çözmek için hibrit bir paradigma sunmuştur (Aljawarneh et al., 2018). Modelin doğruluğunu artırmak, en alakalı özellikleri bulmak için bilgi kazancı ve oylama tekniği kullanılırken, sınıflandırma algoritmaları olarak Saf Bayes, AdaBoostM1, Azaltılmış Hata Budama Ağacı (REPTree), RF, J48, meta sayfalama (meta pagging) ve karar kütüğü (Decision Stump) uygulanmıştır. NSL-KDD veri kümesi üzerinde model değerlendirilmiştir. Bulgular, doğruluğun arttığını ve yanlış

pozitiflerin oranının azaldığını ortaya koymuştur. Önerilen teknik, farklı optimizasyon stratejileri ile diğer veri kümelerinde kullanılarak bu çalışma daha da geliştirilebilir.

Adhi Tama ve diğerleri (2019)'nin çalışmasında iki seviyeli sınıflandırıcı topluluklarına dayalı bir özellik seçimi yöntemi geliştirilmiştir (Adhi Tama et al., 2019). Eğitim veri kümelerinin özellik boyutu karınca kolonisi algoritması, parçacık sürüsü optimizasyonu ve genetik algoritma teknikleri kullanılarak azaltılmıştır. Sınıflandırma performansını en üst düzeye çıkaran öznitelikleri seçmek için REPTree kullanılmıştır. Ardından, rotasyon ormanı (Rotation Forest) ve torbalama yaklaşımları olarak adlandırılan iki seviyeli bir sınıflandırıcı koleksiyonu kullanılmıştır. NSL-KDD ve UNSW-NB15 veri kümeleri kullanılarak önerilen sistem değerlendirilmiştir. Yazarlara göre önerilen model, NSL-KDD, UNSW-NB15 veri setlerinde sırasıyla %85,8 ve %91,3 doğruluk oranlarıyla diğer tekniklerden daha iyi performans göstermiştir. Bu çalışma, daha iyi bir doğruluk üretirken daha az özellik kullanan yeni tekniklerle geliştirilebilir.

Dwivedi ve diğerleri (2019) daha iyi bir STS modeli için, Özellik Seçimi Topluluğu (EFS) ve Uyarlanabilir Çekirge Optimizasyon Algoritması (AGOA) tekniklerini birleştirmiştir (Dwivedi et al., 2019). Sonuçlara göre, %99,23'lük bir tespit oranı artışı, %99,13' lük bir doğruluk ve %0,067'lik düşük bir yanlış alarm oranı elde edilmiştir.

Y. Yang (2019) bilinmeyen izinsiz girişleri tespit etmek için AE ve DNN uygulamıştır (Y. Yang et al., 2019). Algoritmalar NSL-KDD veri kümesi üzerinde test edilmiştir. Model, hafiflik ve düşük kaynak tüketimi açısından daha iyi bir performans göstermiştir; ancak, düşük tespit doğruluğuna ve yüksek eğitim süresine sahiptir.

Meftah ve diğerleri (2019), UNSW-NB15 veri setini kullanarak iki aşamalı anomali tabanlı bir STS tekniği oluşturmuştur (Meftah et al., 2019). En önemli özellikleri seçmek için özyinelemeli özellik eleme ve rastgele ormanlar kullanılmıştır. Daha sonra ikili sınıflandırmayı gerçekleştirmek ve anormal trafiği bulmak için destek vektör makinesi, gradyan artırma makinesi ve lojistik regresyon kullanılmıştır. SVM ile %82,1 doğruluk elde edilmiştir. Sonuçlara göre, iki aşamalı hibrit sınıflandırmanın kullanılması %86,04'lük bir doğruluk artışı sağlamıştır.

Gu ve diğerleri (2019) tarafından özellik artırmalı SVM topluluğu, izinsiz giriş tespitini gerçekleştirmek için kullanılmıştır (Gu et al., 2019). Orijinal özellikler, özellikle yeni ve üstün bir değiştirilmiş eğitim seti oluşturmak için başlangıçta

logaritma marjinal yoğunluk oranları kullanılarak dönüştürülmüştür. Saldırı tespit modeli daha sonra SVM topluluğu kullanılarak oluşturulmuştur. Deneylerin bulguları, önerilen modelin %99,91 performans doğruluğuna, %99,92 tespit oranına ve %0,11 yanlış pozitif oranına sahip olduğunu göstermiştir.

Kanimozhi ve Jacob (2019) tarafından CSE-CIC-IDS2018 veri setine Yapay Sinir Ağları (ANN) uygulandığında benzer sonuçlar elde edilmiştir (Kanimozhi & Jacob, 2019). Düşük yanlış alarm oranına sahipken %99,9 doğruluk, geri çağırma ve kesinlik elde edilmiştir. Ancak model yalnızca birkaç farklı saldırı türünü dikkate almakta; eğitim ve ön işleme için çok fazla zaman harcamaktadır.

Zhao ve diğerleri izinsiz girişlerin tespitinde yüksek yanlış pozitif ve zayıf tespit oranı sorununu ele almak amacıyla çok katmanlı algılayıcı sinir ağlarını (MLP-NN) ve çok amaçlı genetik algoritmaları (MOGA) kullanan bir STS modeli geliştirmiştir (Zhao et al., 2019). NSL-KDD veri seti için önerilen yöntem, %97 doğruluk ve %2'lik yanlış pozitif oranı üretmiştir. Yaklaşımın en büyük dezavantajı uzun zaman alması ve yinelemeli hesaplamalar gerektirmesidir. Stratejinin uygulanabilirliğini göstermek için daha gerçekçi bir veri seti dikkate alınmalıdır. Çünkü bu aynı zamanda veri setinin küçük alt kümeleri kullanılarak da test edilmiştir.

Gao ve diğerleri (2019), çoklu DT'yi uyarlanabilir bir oylama mekanizması ile birleştiren uyarlanabilir bir topluluk öğrenme modeli sunmuştur (Gao et al., 2019). Önerilen model NSL-KDD veri kümesi kullanılarak değerlendirilmiştir. Sonuçlara göre, uyarlanabilir doğruluk %84,23 ile %85,2 arasındadır.

Y. Xiao ve diğerleri (2019) Anomali tabanlı STS için yeni bir Derin Öğrenme (DL) modeli sunmuştur (Y. Xiao et al., 2019). Modeli oluşturmak için KDD CUP 99 veri kümesi ve CNN algoritması kullanılmıştır. PCA ve Autoencoder kullanılarak verilerin boyutluluğunun azaltılması, CNN giriş katmanına verilen değiştirilmiş verileri üretmiştir. Ağ trafiğinin özellikleri daha sonra CNN modeli kullanılarak çıkarılmış ve incelenmiştir. Toplamda %94 doğruluk oranına ulaşılmıştır. Ancak U2R ve R2L saldırı türleri düşük tespit oranlarına sahip olup, saldırıların sırasıyla %20,61 ve %18,96'sı tespit edilmiştir.

Yuyang Zhou ve diğerleri (2020) tarafından, özellik seçimi ve topluluk öğrenme yöntemlerinin dayanan yeni bir STS mimarisi önerilmiştir (Yuyang Zhou et al., 2020). En iyi özellik alt kümesini seçmek için, Korelasyon Tabanlı Özellik Seçimi Yarasa

Algoritması (CFS-BA) kullanmıştır. Daha sonra, saldırı tespiti için RF ve C4.5 entegre edilmiştir. CIC-IDS-2017, AWID ve NSL-KDD veri setleri kullanılarak model test edilmiştir. Sonuçlar, diğer yaklaşımlarla karşılaştırıldığında, önerilen algoritmanın %99,81'lik daha yüksek bir performans doğruluğuna sahip olduğunu göstermiştir.

Swarna Priya ve diğerleri (2020), Kaggle veri kümesinden Gri Kurt Optimizasyonu ile Temel Bileşen Analizi (PCA-GWO) ve Otomatik Kodlayıcı (AE) sınıflandırıcısını kullanan hibrit bir strateji önermiştir. Önerilen model %99,9 doğruluk oranına, %95,4 hassasiyet (sensitivity) oranına ve %100 özgüllük (specificity) oranına ulaşmıştır (Swarna Priya et al., 2020). Ancak çok sınıflı sınıflandırma dikkate alınmamıştır ve model kaynak gerektirmektedir.

Kim ve diğerleri (2020), CSE-CIC-IDS2018 veri kümesinde bir CNN tekniği kullanarak %99 doğruluk elde eden bir model önermiştir. Ancak yöntem yalnızca hizmet reddi saldırılarına odaklanıp, kaynak tüketmiştir (Kim et al., 2020).

Elmasry ve diğerleri (2020), iki katlı bir Parçacık Sürüsü Optimizasyonu (PSO) tekniği önermiştir (Elmasry et al., 2020). Bu teknik, ön eğitim aşamasında optimum özellik alt kümesini ve hiper parametreleri otomatik olarak seçmek için kullanılmıştır. CICIDS2017 ve NSL-KDD veri kümelerinde model değerlendirilmiştir. Derin Sinir Ağları (DNN), Uzun Kısa Süreli Bellek Tekrarlayan Sinir Ağları (LSTM-RNN) ve Derin İnanç Ağları olarak üç farklı derin öğrenme tekniği kullanılarak performans farklılıkları değerlendirilmiştir. Sonucun analizi, önerilen modelin ağdaki izinsiz girişlerin tespitinde dikkate değer bir gelişme sağladığını göstermiştir. Önerilen model, aynı veri seti üzerinde ön eğitim olmadan; aynı modellerin karşılaştırılabilir değerlerine kıyasla algılama oranında %4'ten %6'ya önemli bir artış ve yanlış alarm oranında %5'den %1'e azalma elde etmiştir. DBN ile birleştirilmiş iki katlı PSO, NSLKDD veri setinde %99,81 ve %0,23 yanlış alarm oranıyla en iyi tespit oranını elde ederken, CICIDS2017 veri setinde %99,92 tespit oranı ve 0,1 yanlış alarm oranı elde etmiştir.

Singh Bhati ve diğerleri (2020) KDD Cup99 veri setinde, ayırıcı sınıflandırıcılar topluluğunu kullanarak izinsiz giriş tespiti için yeni bir model sunmuştur (Singh Bhati et al., 2020). Test ve eğitim, rastgele alt uzay algoritması kullanılarak gerçekleştirilmiştir. Sonuç, önceki yöntemlerle karşılaştırıldığında modelin DoS,

Probing, R2L ve U2R için tüm saldırı sınıflarını tespit etmede %98,9 genel doğrulukla daha iyi tespit doğruluğu elde ettiğini göstermiştir.

J. Zhang ve diğerleri (2020), uzun kısa süreli belleği ve çok ölçekli evrişimli sinir ağını (MSCNN-LSTM) birleştiren kapsamlı bir model geliştirmiştir (J. Zhang et al., 2020). Veri setinin mekansal ve zamansal yönleri sırasıyla MSCNN tarafından analiz edilip ve LSTM tarafından işlenmiştir. Sınıflandırmayı gerçekleştirmek için model, mekansal-zamansal özellikleri kullanmıştır. UNSWNB15 veri seti üzerinde sistem değerlendirilmiştir. MSCNN-LSTM modeli, deneysel bulgulara göre %95,6 doğruluk, %1,6 yanlış negatif oranı (FNR) ve %9,8 yanlış alarm oranına sahip olarak geleneksel sinir ağlarından daha iyi performans göstermiştir. Ancak diğerleriyle karşılaştırıldığında yüksek hesaplama süresi sergilemiştir. Ayrıca model nadir görülen saldırıları dikkate almadığı gibi veri kümelerindeki dengesizlik sorununu da çözmeye çalışmamıştır.

Xukui Li ve diğerleri (2020), STS için Otomatik Kodlayıcı Saldırı Tespit Sistemi yaklaşımını önermiştir (Xukui Li et al., 2020). Model, etiketli bir veri kümesi yerine normal trafik kullanılarak eğitilmiştir. Yöntem, en önemli özelliklerin etkili bir şekilde seçilmesi için rastgele orman kullanıp daha sonra seçilen özellikler, özellik gruplandırmayı gerçekleştirmek için benzeşim yayılımı (AP) algoritması kullanarak birden fazla alt kümeye sınıflandırılmıştır. Son olarak ağ trafiğinde anormallik tespiti için AE, K-Ortalamalar ve Gauss Karışım Modeli (GMM) uygulanmıştır. AE, gelen trafiği yönetir ve ardından değeri ağ trafiğinin normal olup olmadığına ilişkin bir fikir verme kriteri olan Karekök Ortalama Hatasını (RMSE) hesaplamıştır. Sınıf dengesizliği sorunu da AE kullanılarak giderilmiştir. Deneysel sonuçlar, çevrimiçi tabanlı AE STS'nin özellik seçimi ve özellik gruplaması yoluyla hesaplama karmaşıklığını azaltabildiğini göstermiştir. Ayrıca diğer geleneksel makine öğrenimi yaklaşımı ve bazı çevrimdışı yöntemlerle karşılaştırıldığında, onlara göre daha iyi bir performans göstermektedir. Ancak ağ saldırılarının mevcut doğası gereği daha şifreli hale gelmiş ve bunun sonucunda normal ağ trafiğine karışmış olduğundan, ağ trafiğinden özellikler çıkarmak iyi bir STS modelini garanti etmek için yeterli değildir. Bu nedenle sistem günlükleri gibi diğer bilgilerin dikkate alınması zorunludur. Ek olarak, model yalnızca geri çağırma ve hesaplama süresi kullanılarak değerlendirilmiştir.

Shahraki ve diğerleri (2020) ikili izinsiz giriş tespit sınıflandırıcıları için uyarlanabilir artırma (AB) çeşitlerinden; Modest AB, Real AB ve Gentle AB etkinliğini incelemişlerdir (Shahraki et al., 2020). Sonuçlar, Gentle AB ve Real AB ile karşılaştırıldığında Modest AB sınıflandırıcısının daha kötü performans gösterdiğini ve daha yüksek hata oranına sahip olduğunu göstermiştir. Gentle ve Real AB'nin her ikisi de hata oranları açısından Modest AB'den yaklaşık %70 daha iyi performans gösterirken, bu onların izinsiz giriş tespiti bağlamında ikili sınıflandırmaya uygunluğunu göstermektedir. Ancak Modest AB, işlem süresi açısından yaklaşık %7 oranında onlardan daha iyi performans göstermiştir. Bununla birlikte, model çoklu sınıf sınıflandırmasını dikkate almamıştır.

Alazzam ve diğerleri (2020), sağlam bir STS oluşturmak için gereken özelliklerin sayısını azaltmayı, aynı zamanda iyi bir doğruluk oranını ve düşük yanlış pozitifleri korumayı hedefleyen, güvercinden ilham alan bir optimize edici (PIO) özellik seçim algoritmasını ortaya koymuştur (Alazzam et al., 2020). UNSW-NB15, NSL-KDD ve KDD CUP 99'daki özelliklerin sayısı, önerilen PIO yöntemiyle sırasıyla 49'dan 5'e, 41'den 5'e ve 41'den 7'ye önemli ölçüde azaltılmıştır. Deneysel sonuçlar ayrıca yüksek doğruluğun ve doğru pozitif oranının (TPR) korunduğunu ve eğitim süresinin kısaltıldığını göstermiştir.

Zhou ve diğerleri (2020), parçacık sürüsü optimizasyonu gibi taktikler kullanarak, birden fazla eğri algoritması altındaki alanla değiştirilmiş uyarlanabilir güçlendirme (MABA) tabanlı sınıflandırıcılarla bir araya getiren bir topluluk sistemi sunmuştur (Ying Zhou et al., 2020). Sistem, sınıf dengesizliği sorununu çözerken ağa izinsiz girişleri etkili bir şekilde tespit etmek için tasarlanmıştır. Model, veri bilgisi kaybını önleyen özellik seçimi veya değişken ölçeklendirme gibi veri ön işlemeyi gerektirmemiştir. Model, %0,00'lık yanlış pozitif oranına ve %0,99'luk bir doğruluğa ulaşmada başarılı olmuştur.

Nguyen ve Kim (2020) ağ kimlikleri için yeni bir algoritma sunmuştur (Nguyen & Kim, 2020). Geliştirilmiş bir özellik alt kümesini seçmek için CNN uygunluk fonksiyonuna sahip GA uygulanırken, seçilen bu özellikleri iyileştirmek için Bulanık C ortalama Kümeleme kullanılmıştır. CNN ayrıca, saldırı karakter türlerinin çoğunu birkaç katman aracılığıyla öğrenme yeteneğine sahip, yüksek kaliteli derin özellik alt kümesi üretmek için bir çıkarıcı olarak kullanılmıştır. Son olarak torbalama (BG), 5 kat çapraz doğrulama kullanılarak doğrulanan bir sınıflandırıcı olarak uygulanmıştır.

Sonuç, GA, FCM ve CNN'nin yüksek kaliteli bir özellik seti üretebildiğini göstermiştir. CNN ve BG'nin hibriti ise önemli ölçüde %98,2'lik gelişmiş bir algılama performansına, %0,5 yanlış alarm oranına ve %95,4'lük doğru pozitif oranına yol açmıştır. Bununla birlikte, önerilen model, ilgili özelliklerin seçimi için uygulanan GA ve CNN'nin bir sonucu olarak çok zaman harcamıştır. Ayrıca, beş kat çapraz doğrulamanın yinelenmesi de pratik simülasyon için çok zaman almıştır.

Z. Wu ve diğerleri (2020) ağ anomali tespit modelinin genelleme kapasitesini önemli ölçüde artırmak için, Artık Ağ (ResNet)'ı semantik yeniden kodlamayla birleştirmiştir (Z. Wu et al., 2020). Kapsamlı anlamsal kodlama alanıyla önerilen model, ağ trafiği anormallik tespitini umut verici bir sonuçla ele almıştır. Bununla birlikte, anlamsal yeniden kodlama teknolojisi, düşük anlamsal alana sahip ağ trafiği için trafik tanımlama performansını yalnızca biraz iyileştirmiştir.

Mebawondu ve diğerleri (2020), ANN tabanlı bir STS modeli önermiştir (Mebawondu et al., 2020). UNSW-NB15 veri seti kullanılmıştır. Sonuç, yöntemin %76,96 doğruluk oranına ulaştığını ortaya çıkarmıştır. Ancak sistemin doğruluğu zayıftır.

Krishnaveni (2020), NSL-KDD veri setlerinden özel özellikler kümesini seçmek için SVM sınıflandırıcısı ve bilgi kazanç oranı (IGR) kullanarak bu iki tekniğe dayanan etkili bir anomali tabanlı STS sunmuştur (Krishnaveni et al., 2020). Eğitim ve saldırı tespiti için SVM kullanılmıştır. Deneysel bulgulara göre; bilgi kazanç oranı ile SVM kullanımı, doğruluğu %96,24'e çıkarmış ve yanlış alarm oranını azaltmıştır.

Ashiku & Dagli (2021) yaptıkları çalışmada, UNSW-NB15 veri seti üzerinde Evrişimli Sinir Ağları (CNN) tekniği kullanılarak STS için bir model geliştirilmiş ve modelin doğruluk oranı %94,4 olmuştur (Ashiku & Dagli, 2021). Ancak model, verilerdeki sınıf dengesizliği sorununu ele almamış; ayrıca tespit oranı önemli ölçüde düşüktür ve sıfırıncı gün saldırısını etkili bir şekilde tespit edememiştir.

Jia ve diğerleri (2021), ağ saldırı tespit sistemleri için KDDCup99 veri kümesini kullanarak bir Bilgi Entropisi Derin İnanç Ağları (IE-DBN) modeli önerilmiştir (Jia et al., 2021). Boyutsallıklarını en aza indirmek ve gereksiz özellikleri silmek için Bilgi Kazancı (IG) kullanılırken, Derin İnanç Ağları (DBN) ağındaki gizli nöronların sayısı ve derinliği Bilgi Entropisi (IE) kullanılarak belirlenmiştir. Veri sınıfı dengesizliği sorununu çözmek için Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE) algoritması da kullanılmıştır. %98,76 algılama doğruluğu ve 0,76 yanlış alarm oranı

(FAR) elde edilmiştir. Bu, DBN ile IE ve IG kullanmanın modelin öğrenme verimliliğini ve algılama doğruluğunu iyileştirdiğini, ağın ve FAR'ın hesaplama maliyetinin azaldığını göstermiştir.

Oche Onah ve diğerleri (2021), anormalliklerin tespiti için NSL-KDD veri kümeleri kullanılarak, Genetik Algoritma Sarmalayıcı Tabanlı özellik seçimi ve Anomali Tespit Modeli için Saf Bayes (GANBADM) önermiştir (Oche Onah et al., 2021). Boyutsallığın azaltılması için sarmalayıcı tabanlı genetik algoritma uygulanırken, indirgenmiş veri kümesi üzerinde sınıflandırma Saf Bayes kullanılarak yapılmıştır. Sonuçlar, önerilen modelin %0,6 yanlış alarm oranına ve %99,73 doğruluğa sahip olduğunu; Rastgele Orman (RF), Destek Vektör Makinesi (SVM) ve Karar Ağacı (DT) gibi diğer sınıflandırıcılardan daha iyi performans gösterdiğini ortaya çıkarmıştır. Ancak f-puanı (F-Score) düşük ve yürütme süresi daha uzundur.

Gu ve Lu (2021) tarafından bir SVM ve Saf Bayes STS çerçevesi sunulmuştur (Gu & Lu, 2021). Yeni, yüksek kaliteli veriler oluşturmak için orijinal özellikler, Saf Bayes tekniği kullanılarak değiştirilmiş ve SVM sınıflandırıcısı, saldırı tespit modelini oluşturmak için değiştirilmiş veriler kullanılarak eğitilmiştir. Deneysel bulgular UNSW-NB15, CICIDS2017, NSL-KDD ve Kyoto 2006+ veri kümeleri için sırasıyla %93,75, %98,92, %99,35 ve %98,58 doğruluk oranlarında iyi bir performans göstermiştir. Ayrıca strateji, tespit oranı ve yanlış alarm oranı açısından diğer tekniklerden daha iyi performans göstermiştir. Saldırı tespiti için sadece ikili durum dikkate alınsa da modelin farklı saldırı türlerine sahip senaryolara genişletilmesine ihtiyaç vardır. Ayrıca metodolojide sınıf dengesizliği gerektiği gibi ele alınmamıştır.

Nazir ve Ahmed Khan (2021) tarafından bir arama yöntemi olarak tabu aramanın (TS) ve STS için öğrenme algoritması olarak rastgele ormanın kullanıldığı bir çalışmada, TS rastgele orman olarak bilinen sarmalayıcı tabanlı bir özellik seçme stratejisi önerilmiştir (Nazir & Ahmed Khan, 2021). Önerilen modeli değerlendirmek için UNSW-NB15 veri seti kullanılmıştır. Daha önce kullanılan Chi-square ve Pearson korelasyon özellik seçme prosedürleriyle karşılaştırıldığında doğruluk ve yanlış pozitif oranları sırasıyla, %83,12 ve %3,7 elde edilmiştir. Ancak veri kümesinin sınıf dengesizliği sorunu ele alınmadığından; sınıflandırıcının doğruluğu üzerinde olumsuz bir etkiye sahip olabilir, yanlış pozitif ve yanlış sınıflandırma oranını artırabilir.

Wang ve diğerleri (2021), Derin İnanç Ağı Softmax'a (DBN Softmax) ve Yığılmış Gürültü Giderici Otomatik Kodlayıcı Aşırı Öğrenme Makinesi' ne (SDAE-ELM) dayalı bir STS modeli önermiştir (Wang et al., 2021). İlk aşamada, veri kümesinin özelliklerini öğrenmek için SDAE uygulanmış ve daha sonra eğitilen modeli üretmek için ELM algoritmasına girdi olarak hizmet etmiştir. Son olarak, saldırı tespiti için test verilerine SDAE-ELM modeli uygulanmıştır. DBN aşamasındaki her katman Kısıtlı Boltzmann Makinesi'dir (RBM). Bu da tüm ağın farklı RBM'lerden oluşan bir yığın olduğu anlamına gelmektedir. Tüm ağı eğitmek için katman katman, denetimsiz bir eğitim süreci kullanılmıştır. Ardından izinsiz girişin türünü belirlemek ve doğruluğu artırmak ve DBN modelindeki Softmax sınıflandırıcısını kullanarak verileri kategorize etmek için geri yayılım (BP) tekniği kullanılmıştır. SDAE ve DBN modellerini optimize etmek için ELM algoritması ve Softmax sınıflandırıcısı kullanılmıştır. Deneysel sonuçlar, her iki modelin de geleneksel makine öğrenimi modellerine göre ikili ve çoklu sınıf sınıflandırmanın her iki düzeyinde de iyi bir algılama doğruluğu ürettiğini göstermiştir. Ancak SDAE-ELM modeli etkin yetenek gösterse de küçük veri seti üzerinde tespit yeteneği zayıftır. Ayrıca DBN Softmax modelinin büyük veri kümeleri üzerinde eğitimi çok fazla zaman almakta ve gerçek zamanlı algılama açısından gerçekçi değildir.

Liu ve diğerleri (2021) uyarlanabilir sentetik (ADASYN) aşırı örnekleme tekniği ve LightGBM topluluğu uygulayarak bir STS modeli önermişlerdir (J. Liu et al., 2021). Veri dengesizliği sorununu çözmek için ADASYN uygulanmıştır. LightGBM topluluğunun uygulanmasının yanı sıra, ADASYN; sınıflandırıcı olarak kullanılmış, modelin eğitimi ve izinsiz giriş tespit süresi ile ilişkili hesaplama karmaşıklığı sorununu ele almıştır. Model UNSW-NB15, NSL-KDD ve CICIDS2017 veri setleri kullanılarak değerlendirilmiştir. Deneysel bulgulara göre LightGBM, üç veri seti için sırasıyla %83,98, %89,79 ve %99,86 doğruluk elde etmiştir. Ayrıca, ADASYN aşırı örnekleme uygulandıktan sonra azınlık sınıflarının tespit oranı iyileştirilmiştir ve böylece sırasıyla %85,89, %92,57 ve %99,91 genel tespit doğruluğu elde edilmiştir. Ancak yanlış pozitiflik oranı nispeten yüksektir; bu nedenle model bu sorunu çözecek şekilde geliştirilebilir.

Thilagam ve Aruna (2021) tarafından, izinsiz giriş tespiti için Tekrarlayan Konveksiyonel Sinir Ağı (RC-NN) önerilmiştir (Thilagam & Aruna, 2021). Modele CNN, LSTM ve Karınca Aslanı Optimizasyon (ALO) algoritmaları dahil edilmiştir.

DARPA ve CSE-CIC-IDS2018 veri kümeleri kullanılarak önerilen strateji değerlendirilmiş ve mevcut yöntemlerle karşılaştırılmıştır. Model, %0,0012'lik azaltılmış hata oranına ve %94'lük artırılmış sınıflandırma doğruluğuna ulaşmıştır. Sonuç olarak bu yöntem, her saldırıyı önceki araştırmalara göre daha doğru bir şekilde tanımlayabilmektedir.

Rajesh Kanna ve Santhi (2021), izinsiz giriş tespiti için hiyerarşik çok ölçekli LSTM (HMLSTM) ve optimize edilmiş CNN'nin (OCNN) birleşik bir modelini sunmuşlardır (Kanna & Santhi, 2021). CNN'nin hiperparametreleri, mekansal bilgilerin öğrenilmesine yönelik en iyi konfigürasyon için Aslan Sürüsü Optimizasyonu (LSO) kullanılarak ayarlanmıştır. HMLSTM zaman özelliklerini toplar ve çeşitli özellikler arasındaki hiyerarşik ilişkileri öğrenir. Bu bilgi daha sonra birleşik model kullanılarak ağ verilerinin daha ileri düzeyde sınıflandırılması için kullanılmıştır. Modeli değerlendirmek için NSL-KDD, ISCX-IDS ve UNSWNB15 veri kümeleri kullanılmıştır. Deneysel bulgular, önerilen modelin alternatif STS tekniklerinden daha iyi performans gösterdiğini ortaya çıkarmıştır. Sonuçlar, OCNN-HMLSTM modelinin çok sayıda saldırıyı etkili bir şekilde tespit ettiğini ve üç veri kümesinin tamamında izinsiz giriş tespitini iyileştirdiğini göstermiştir.

Sona ve Sasirekha (2021), Kulczynski Benzerlik İndeksli Yusufçuk Optimizasyonuna Dayalı Çok Kategorili Uyarlanabilir Temel Sınıf Topluluğu (KSIDO-PABCE) anlamına gelen KSIDO-PABCE'yi temel alan bir STS modeli önermiştir (Sona & Sasirekha, 2021). Bu modelin daha az zaman harcayarak daha fazla doğruluk sunması amaçlanmaktadır. Özellikleri seçmek için KSIDO kullanılmıştır. Arama alanındaki yusufçukların veya özelliklerinin başlangıç popülasyonu, optimizasyon süreci yoluyla üretilir. Kulczynski benzerlik indeksi, her bir yusufçuğun başlangıçtan sonraki uyum durumunu değerlendirmek için kullanılır. En iyi özellikler uygunluk değerlendirmesine göre seçilmiştir. Seçilen ideal niteliklere sahip çeşitli saldırı türlerini keşfetmek için PABCE tekniği kullanılmıştır. Önerilen yöntem, farklı saldırı türlerinin tespit doğruluğunu minimum sürede artırmıştır.

Alazzam ve diğerleri (2021) tarafından, iki temel paralel çalışan alt sisteme sahip hafif bir ağ tabanlı STS paradigması önerilmiştir (Alazzam et al., 2021). Her alt sistem, tek sınıf destek vektör makinesi (OCSVM) algoritması kullanılarak eğitilmiştir. İkinci alt sistem saldırı paketleri üzerinde eğitilirken, birinci alt sistem normal paketler üzerinde eğitilmiştir. Ağ boyunca hareket eden her paketin sağlam bir değerlendirmesini

sağlamak için iki alt sistemin sonuçları birlikte kullanılmıştır. KDDCUP-99, NSL-KDD ve UNSW-NB15 veri setlerinde önerilen model değerlendirilmiştir. Bulgular, önerilen ağ tabanlı STS modelinin literatürde halihazırda kullanımda olan alternatif yaklaşımlardan daha iyi performans göstermiştir.

Murtugudde (2021), genetik tabanlı geliştirilmiş Gri Kurt Optimizasyonu (GB-EGWO) sunmuştur (Murtugudde, 2021). Genetik çaprazlama tekniği GB-EGWO tarafından kullanılmıştır ve ağ saldırılarını tanımlamak için en iyi çözümü bulmak amacıyla konumu hareket ettirip hareket ettirmeyeceğine karar vermeden önce en iyi arama ajanı (çözüm) gerçekleşene kadar özellik seçimi sürecinin tekrarlanmasını içermektedir. Model NSL-KDD veri seti üzerinde değerlendirilmiştir. GB-EGWO algoritması %98,62'lik gelişmiş saldırı tespit doğruluğu göstermiştir. Ancak veri kümesindeki saldırıyı tanımlamak için gereken süreyi azaltmak önemlidir.

Rao ve diğerleri (2021) STS için iki aşamalı bir hibrit model tanıtmıştır (Rao et al., 2021). İlk olarak, seyrek otomatik kodlayıcı (SAE) ile yumuşatılmış 11 düzenlemesi (Smoothed 11 Regularization) kullanılmıştır. İkincisi, saldırılar Derin Sinir Ağı (DNN) kullanılarak tahmin ve kategorize edilmiştir. UNSW-NB15 veri setinde deneysel bulgular, sunulan tekniğin %99,98 doğruluk ve %99,99 tespit oranıyla diğer geleneksel tekniklerden daha iyi performans gösterdiğini bildirmiştir. Ancak yöntem, özellik öğrenme ve boyutluluk azaltma süreçlerini daha iyi temsil etmek için çeşitli otomatik kodlayıcılar kullanılarak geliştirilebilir.

Sharma ve Yadav (2021), saldırıları tanımlamak için bir dizi makine öğrenme tekniği (karar ağacı, destek vektör makinesi, rastgele orman ve diskriminant analizi biçiminde bir topluluk sınıflandırıcı) ile birlikte Özyinelemeli Özellik Eleme (RFE) tekniğini kullanmışlardır (Sharma & Singh Yadav, 2021). RFE, KDD CUP 99 veri kümesindeki gereksiz özellikleri ortadan kaldırmak için kullanılmış ve veri kümesinin boyutsallığının azaltılması açısından önemlidir. Deneylerin sonuçları, önerilen yaklaşımın tüm saldırı sınıfları için iyi sınıflandırma oranları ürettiğini göstermiştir. Özellik seçiminden önce, üç sınıflandırma yönteminin karşılaştırılması, rastgele ormanın SVM'den daha iyi performans gösterdiğini ortaya çıkarmıştır. Bununla birlikte, özellik seçiminden sonra SVM, rastgele orman ve karar ağacından daha iyi performans göstermiştir.

Halim ve diğerleri (2021), sınıflandırıcıların kesinliğini (presicion) artırmak için GA tabanlı özellik seçimi (GbFS) tekniğini önermiştir (Halim et al., 2021). CIRA-CIC-DOHBrw-2020, UNSW-NB15 ve Bot-IoT veri setleri kullanılarak, geliştirilmiş GA tabanlı özellik seçimi yaklaşımı test edilmiştir. %99,80 doğruluk oranına sahip olan GbFS, geleneksel özellik seçme teknikleriyle karşılaştırıldığında daha doğru performans göstermiştir. Bununla birlikte, çok sayıda tekrarlanan tur ve çoğaltma işlemleri nedeniyle GA'nın uygulanması, temelde yavaş çalışmaya yol açmıştır. Ek olarak, önerilen sistem tahmin aşamasında üç denetimli öğrenme sınıfı kullanır; dolayısıyla, makinenin yeni saldırı türlerini kendi başına öğrenmesini sağlamak için kümeleme gibi denetimsiz bir öğrenme tekniği uygulanabilir.

Xin Li ve diğerleri (2021), STS'lerin düşük performansını ve yüksek yanlış pozitif oranlarını ele almak için, Kril Sürüsü Doğrusal En Yakın Komşu Lasso Adım Optimizasyonuna (LNNLS-KH) dayanan gelişmiş bir Krill Sürüsü Tekniği (Krill Swarm Technique) sunmuştur (Xin Li et al., 2021). Deneysel bulgulara göre, LNNLS-KH algoritması NSL-KDD ve CICIDS2017 veri kümelerinden gereksiz özellikleri silerken, her veri kümesinden sırasıyla 7 ve 10 anlamlı özellik seçmiştir. Elde edilen sonuçlara göre, önerilen strateji saldırı tespit doğruluğunu artırmış ve ayrıca yakınsama hızı ve optimum uygunluk yineleme eğrisi açısından iyi performans göstermiştir. Ayrıca düşük bir yanlış pozitif oranına sahiptir.

Gupta ve diğerleri (2021), geliştirilmiş bire karşı bir (I-OVO) tekniğine ve LSTM sınıflandırıcısına dayanan anomali tabanlı STS modelini önermişlerdir (Gupta et al., 2021). Önerilen STS iki katmana ayrılmıştır. LSTM sınıflandırıcı ilk katmanda izinsiz girişleri normal ağ trafiğinden ayırt etmek için kullanılırken, ikinci katmanda tanımlanan izinsiz girişleri farklı saldırı türlerine ayırmak için rastgele orman ve torbalama topluluğu algoritmaları (Bagging Ensemble Algorithms) kullanılmıştır. Model ayrıca verilerdeki sınıf dengesizliğini gidermek için sınırda SMOTE, rastgele aşırı örnekleme ve SVM SMOTE yaklaşımlarını kullanmıştır. Modeli değerlendirmek için NSL-KDD, CICIDS2017 ve CIDDS-001 veri kümeleri kullanılmıştır. NSL-KDD veri setinde model %87 ve %91 doğruluk ve tespit oranına sahipken, CICIDS2017 veri setinde %96 ve %99 doğruluk ve tespit oranına sahiptir. Sonuçlar, önerilen stratejinin izinsiz girişleri zamanında ve doğru bir şekilde tespit ettiğini göstermiştir.

Chiche ve Meshesha (2021), STS performansını artırmak için yeni bir strateji önermiştir (Chiche & Meshesha, 2021). Model, sınıflandırıcı olarak rastgele orman

kullanılarak geliştirilmiştir. Önerilen yaklaşımı uygulamak için NSL-KDD veri kümesi kullanılmış ve sonuçlar %99,91'lik bir doğruluk ortaya koymuştur.

Lv ve diğerleri (2020), kötüye kullanım saldırı tespiti için, hibrit çekirdek fonksiyonuna sahip bir aşırı öğrenme makinesi (HKELM) sunmuştur (Lv et al., 2020). Yerçekimsel arama algoritması (GSA) ve diferansiyel evrim algoritması (DEA) birleştirilerek HKELM'nin parametreleri değiştirilmiş, bu da saldırı tahmini sırasında hem küresel hem de yerel optimizasyon yeteneklerini geliştirmiştir. Ek olarak, özellik çıkarma ve boyut azaltma işlemi çekirdek temel bileşen analizi algoritması (KPCA) kullanılarak gerçekleştirilmiştir. Son olarak, KPCA-DEGSA-HKELM yeni STS tekniği keşfedilmiştir. KDD CUP 99 veri kümesi, UNSW-NB15 veri kümesi ve Tennessee Eastman sürecinden elde edilen veri kümesi üzerinde önerilen teknik test edilmiştir. Deneysel bulgular, önerilen yöntemin mükemmel doğruluğunun yanı sıra zaman tasarrufu avantajlarını da desteklemektedir.

Disha ve Vahid (2022), ağlara izinsiz girişi tespit etmeye yönelik ikili sınıflandırma işi için, karar ağacı, çok katmanlı algılayıcı (MLP), AdaBoost, Gradyan Güçlendirme Ağacı (GBT), Geçitli Tekrarlayan Birim (GRU) ve LSTM dahil olmak üzere çeşitli makine öğrenmesi modelleri kullanılmıştır (Abedin Disha & Waheed, 2022). UNSW-NB 15 ve Network TON IoT (Nesnelerin İnterneti)'den alınan veri setleri kullanılarak modellerin performansı değerlendirilmiştir. UNSW-NB15 veri setinden yirmi önemli özellik ve Network TON IoT veri setinden 10 özellik, Gini Safsızlık Tabanlı Ağırlıklı Rastgele Orman (GIWRF) olarak bilinen entegre özellik seçim tekniği kullanılarak seçilmiştir. Öğrenme algoritmasının sınıf dağılımını kavramasına yardımcı olmak için GIWRF, ağaçları bölme kriteri olarak Gini safsızlığını kullanmış ve dengesiz verilerin iki farklı sınıfı için ağırlıkları ayarlamıştır. Deneysel bulgular, DT'nin özellik seçme yöntemini kullanarak diğer makine öğrenmesi modellerine göre daha iyi performans göstermiştir. Ancak bu çalışma çok sınıflı sınıflandırmayı veya zamansal karmaşıklık analizlerini dikkate almamıştır.

Ahmed ve diğerleri (2022) UNSW-NB15 veri setini kullanarak, farklı ağ saldırı türlerini tanımlamak için rastgele orman, karar ağacı, lojistik regresyon (LR), KNN ve ANN dahil olmak üzere beş makine öğrenmesi tekniğini araştırmıştır (Ahmed et al., 2022). Sınıflandırıcılardan biri olan RF %89,29 ile en iyi doğruluğa sahip olmuştur. Ayrıca, sınıf dengesizliği sorununu ele almak için SMOTE kullanıldıktan sonra sınıflandırma modeli doğruluğunda daha fazla ilerleme kaydedilmiş ve RF

sınıflandırıcısı, temel bileşen analizi yönteminden seçilen 24 özellik ile %95,1'lik en yüksek doğruluğa ulaşmıştır. Ek olarak, sınıf dengeleme LR ve YSA sınıflandırıcıları üzerinde yararlı bir etki göstermemiş, bunun yerine azınlık sınıflarını ele aldıktan sonra doğruluklarını azaltmıştır.

I. Ahmad ve diğerleri (2022) ağ saldırı tespiti için; UNSW-NB 15 veri kümesini kullanarak, AdaBoost tabanlı bir yöntem önermiştir (I. Ahmad et al., 2022). MLP ve SVM teknikleri, önerilen model ile karşılaştırılmıştır. Bulgular, önerilen yöntemin %99,3'lük bir doğruluğa ulaştığını ve ayrıca çeşitli ağ saldırı türlerini tanımlamanın mümkün olduğunu göstermiştir. Qureshi ve diğerleri (2018) benzer şekilde, NSL-KDD veri kümesini kullanarak, benzersiz bir rastgele sinir ağı tabanlı STS önermiş ve %94,5'lik bir doğruluk elde etmiştir (Qureshi et al., 2018).

3. SALDIRI TESPİT SİSTEMLERİNİN SINIFLANDIRILMASI

Anomali veya anormallik, başka bir araç veya kişi tarafından oluşturulduğuna dair şüphe uyandıracak kadar mevcut gözlemlerden sapan bir gözlem olarak ifade edilebilir (Gupta et al., 2016). Aynı zamanda gizlilik, bütünlük ve kimlik doğrulama olan bilgisayar güvenlik modelinin öngördüğü şekilde ağ ve sistemlerin güvenliğini bozan bir eylemdir. Anomali, güvenlik mekanizmasını ihlal ederek, yetkisiz erişim sağlayarak ve ağ içinde veya dışında saldırılar gerçekleştirerek elde edilir. STS, çeşitli kaynaklardan gelen verileri izleyerek bu tür engellere karşı güvenlik sağlayan sistemdir. STS tarafından tespit edilen yaygın saldırı türleri (Daş et al., 2015), açıklamaları ve örnekleri Tablo 1'de listelenmiştir.

STS'nin temel işlevleri aşağıdaki gibidir(Vasilomanolakis et al., 2015):

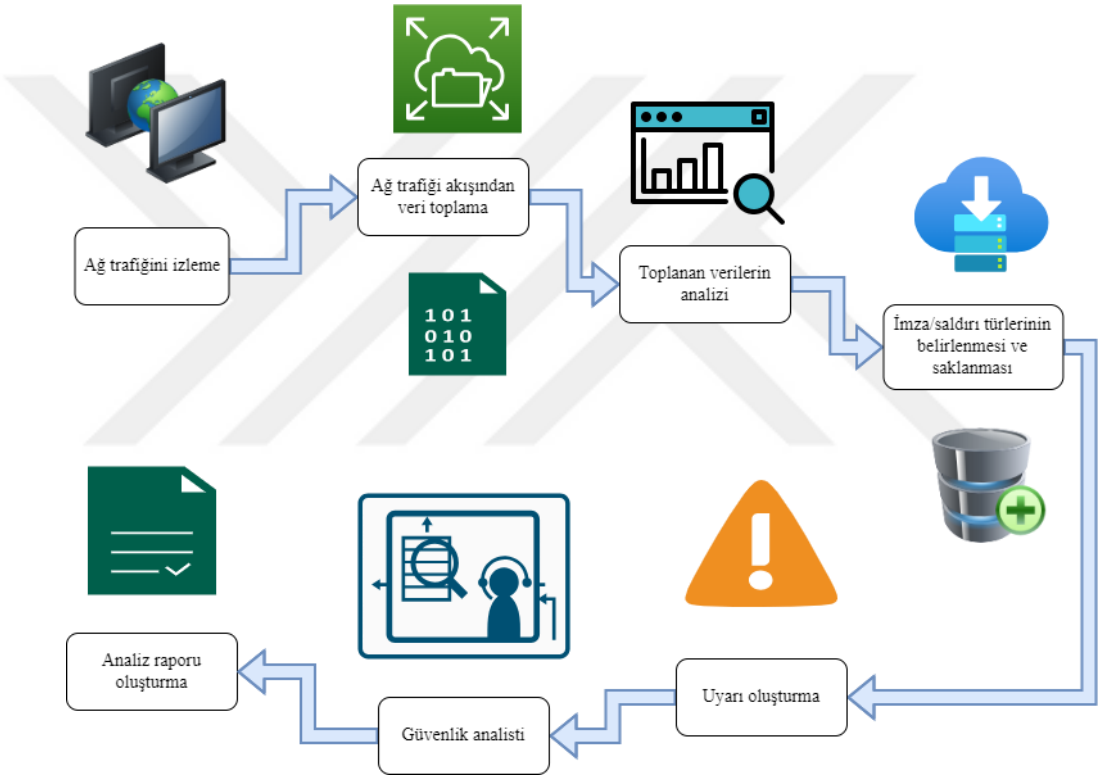
- Ağ trafiğini izleme,
- Ağ trafiğinden toplanan verileri analiz etme,
- Saldırı türlerini veya imzaları tanımlama,
- Saldırı türleri veya imzalarını veri tabanında depolamak,
- Herhangi bir anormallik veya imza eşleşirse uyarı oluşturmak.

Tablo 3.1: STS Tarafından Tespit Edilen Yaygın Saldırı Türleri.

Saldırılar	Örnekler	Tanım
Yönlendirme Saldırıları	Kara Delik Saldırısı, Gri Delik Saldırısı, Solucan Deliği Saldırısı, Sybil Saldırısı, Ortadaki Adam Saldırısı, Bizans Saldırısı	Bunlar, yönlendirme bilgilerini sızdıran, değiştiren veya yeniden yürüten ağ katmanı saldırılarıdır.
Sniffer Saldırıları	Kimlik Avı, İstenmeyen Posta Algılama, Çevrimiçi Dolandırıcılık, Yasa Dışı işlem, Hesap Ele Geçirme, Tahrifat	Bu tür saldırılar, ağ koklama araçları kullanılarak ağ trafiği verilerinin çalınması veya ele geçirilmesi yoluyla gerçekleştirilir.
Pasif Saldırılar	Dinleme, Trafik Analizi	Saldırganın ilgili sistemlerle iletişim kurmadığı ancak aralarındaki veri akışını gözlemleyerek sistemi bozmaya çalıştığı kriptografik sistem üzerinde gerçekleştirilen saldırılardır.
İçeriden Saldırılar	Kullanıcıdan Root' a, Flooding Saldırıları, Port Tarama	Saldırganın sisteme ait olduğu ve ağa erişim yetkisinin verildiği sistem ağına yapılan kötü niyetli saldırılardır. Ayrıca sistemin mimarisi hakkında bilgi sahibi olabilir.
Kötü Amaçlı Saldırılar	Botnet, Kötü Amaçlı Yazılım	Kullanıcının sisteminden yararlanmak için bilgisayar virüsleri gibi kötü amaçlı yazılımlar enjekte ederek veya sosyal mühendislik yaparak sistemi kötüye kullanma yolu olarak tanımlanmaktadır.
DoS/DDoS	Tampon taşması, Ping of Death, ICMP, Smurf, UDP Flood, SYN Flood	Bu saldırılar, meşru kullanıcıların ağ kaynaklarına erişmesini engelleyen, ağ paketlerle dolup, ağı meşgul eden saldırılardır.
Siber Saldırılar	Siber Zorbalık, Siber Casusluk	Gizli, fikri, kurumsal varlıklara zarar vermek veya maddi kazanç sağlamak amacıyla belirli bir kurum ve kişilere yönelik hedefli saldırılardır.
Güvenlik açıkları	Yanlış Yapılandırma, Sistem Açıkları	Bunlar, sistemin tehlikeye atılması için bir olasılık yaratan kod veya sistem tasarımındaki boşluklardır. Bunlar, bir saldırganın kodunu çalıştırabileceği veya bir sistemin belleğine girebileceği olası saldırı noktaları oluşturan kusurlardır.
Diğerleri	Parola Saldırıları, Kaba kuvvet, Sözlük Saldırıları	Bu saldırılar, kullanıcıların şifrelerini veya diğer fikri ve gizli bilgilerini çalmak için gerçekleştirilir.

Daş ve diğerleri (2015)'den uyarlanmıştır.

STS tarafından sağlanan işlevler temelinde, STS'nin bileşenleri Şekil 3.1'de gösterilmektedir. Ağ izlenerek, ağ paketlerinden bilgi toplanır. Saldırganlar, bilgi elde etmek için kötü amaçlı kod enjekte ederek veya ağ paketlerini analiz ederek ağ saldırıları gerçekleştirir. Saldırılar, tüm ağ işlemlerini gerçekleştiren sunucuda veya ağ etkinliklerini gerçekten gerçekleştiren sistem ana bilgisayarında gerçekleşebilir. Sistemde bulunan güvenlik açıklarından yararlanmak için eylemler de gerçekleştirilebilir. Aslında, makine öğrenmesi ve derin öğrenme gibi teknikler, ağ tehditlerini tespit etmek için daha akıllı STS'ye sahip olmamızı sağlar.



Şekil 3.1: STS Bileşenleri.

Vasilomanolakis (2015)'den uyarlanmıştır.

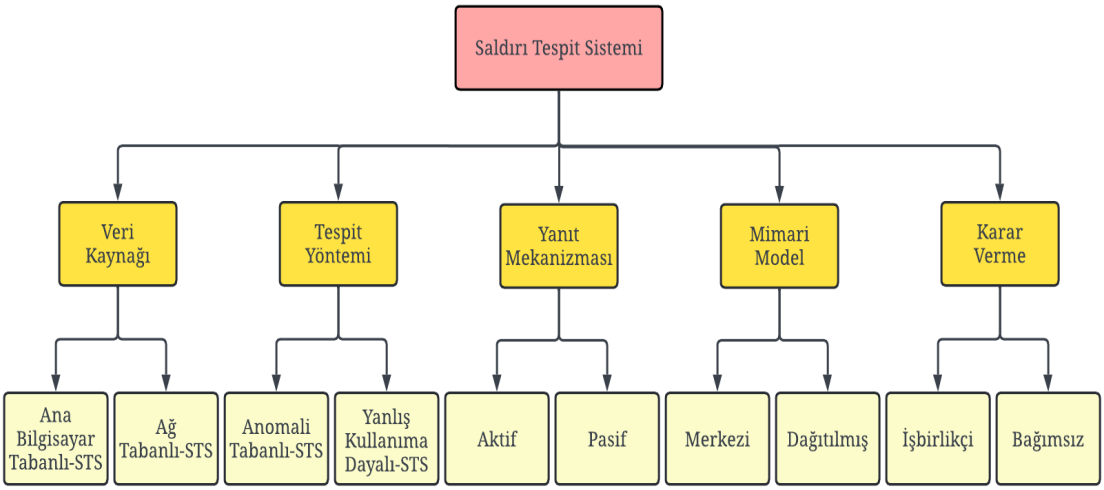
STS'nin bileşenleri aşağıdaki gibidir.

- **Ağı Trafiğini İzleme:** Ağla ilgili bilgileri içeren gerekli paketleri toplamak için bir ağın izlenmesi gerekir. Bir ağ paketi, paket başlığı ve paket yükünün bir birleşimidir. Hem başlık hem de yük, bir saldırı gerçekleştirmek için gerekli bilgileri çıkarmak için faydalı olabilir. Ağ akışı bile, bir saldırı gerçekleştirmek için istismar edilecek veri modellerini bulmak için analiz edilir. Bu nedenle,

izinsiz giriş tespiti için oluşturulan veri kümeleri, saldırıları sınıflandırmak için paket seviyesi ve akış seviyesi özelliklerine sahiptir.

- **Veri Toplama:** Saldırının gerçekleştirileceği hedef sistem hakkında detayların toplanması anlamına gelir. Bu, ağ komutu veya araçları kullanılarak sorgular gerçekleştirilebilir. Örneğin, paket düzeyindeki ayrıntılar, "Wireshark" aracı kullanılarak ağ üzerinden akan paketleri koklayarak veya "nslookup" gibi ağ komutlarını kullanarak alan adı gibi sunucu ve ana bilgisayarla ilgili ayrıntılar elde edilebilir.
- **Verilerin Analizi:** Bu, gizli bilgilerin çalınması için ağ paketinin taranması olarak ifade edilebilir. Örneğin, sistemi tehlikeye atarak ve sisteme yetkisiz erişim elde ederek bir R2L (Uzaktan Yerele) saldırısı gerçekleştirilebilir. Erişim elde etmek için gerçekleştirilebilecek saldırılardan bazıları, sisteme uzaktan erişim sağlamak için paketi koklamak ve kimlik bilgilerini çalmak veya truva atı gibi kötü amaçlı yazılımları enjekte etmektir. Bu tür güvenlik açıklıkları yalnızca hedef sistemde birkaç açık bağlantı noktası varsa kullanılabilir.
- **İmza/Saldırı Türlerinin Belirlenmesi ve Saklanması:** Paket ayrıntılarının analizinden sonraki adım, halihazırda bilinen saldırıların ve yeni saldırıların saldırı modellerini veya içeriden saldırıları başlatmak için kullanılabilecek bazı bilinen açıklardan yararlanmaların imzasını belirlemektir. Bu imzalar ve modeller, gelecekte referans olması için veri tabanında saklanır ve bu nedenle, güvenlik yöneticisi, anormallik bulursa, müdahaleci davranışları kolayca bildirebilir.
- **Uyarı Oluşturma:** Saldırı düzenini tanıdıktan sonra, bir uyarı/alarm oluşturulur ve güvenlik yöneticisine bildirilir. Uyarı, imza/desen eşleşmesine göre tetiklenir.

STS'nin sınıflandırılması Şekil 3.2'de sunulmaktadır. STS tarafından ağın analizi için kullanılan veri kaynağına veya STS tarafından izinsiz girişleri sınıflandırmak için benimsenen tespit yöntemine göre sınıflandırılır. Ayrıca, STS tarafından verilen yanıtın türü, benimsenen mimari ve STS tarafından mimariye dayalı olarak verilen kararın türü de sınıflandırma için nitelikler olarak kabul edilebilir.



Şekil 3.2: STS'nin Sınıflandırılması.

Anwar ve diğerleri (2017), Gupta ve diğerleri (2016), Inayat ve diğerleri (2016), Snapp ve diğerleri (1991), Vidal ve diğerleri (2020)'den uyarlanmıştır.

3.1. Veri Kaynağı

İzinsiz giriş analizi için toplanan veriler çeşitli kaynaklardan toplanabilir ve bu nedenle bilgi kaynaklarına bağlı olarak STS, ana bilgisayar tabanlı STS ve ağ tabanlı STS olarak sınıflandırılır (B. Gupta et al., 2016; Vidal et al., 2020).

3.1.1. Ana Bilgisayar Tabanlı STS

Burada, yerel sisteme veya ana bilgisayara bir STS kurulur. Konfigüre edilmiş ana bilgisayarın denetim izleri, sistemin davranışının durumu, herhangi bir kötü niyetli faaliyetin imzası hakkında bilgi toplamak için incelenir ve yerel sistemi korumak için her türlü önleyici tedbiri alabilir. Denetim izleri, sistem günlükleri, uygulama günlükleri ve ana bilgisayar izleme gibi farklı kaynaklardan bulunabilir. Bu günlükler, işletim sisteminin ağ varlık günlüklerinden, güvenlik duvarı gibi güvenlik mekanizmalarından, yönlendirici ve web sunucusu gibi ağ cihazlarından ve FTP gibi

ağ protokollerinden toplanabilir. Dosya verilerinin tahrif edilmesi, segmentasyon hatası, sistem yazılımının çökmesi, sisteme yetkisiz erişim gibi kötü niyetli faaliyetler kaydedilebilir.

3.1.2. Ağ Tabanlı STS

Burada STS, ağ içindeki ve dışındaki faaliyetleri izlemek için tüm ağ ortamını dikkate alır. Ağ ortamının içinde ve dışında bulunan tüm paketler incelenir. İnceleme için düşünülen ağ trafik verileri, ağı tehlikeye atabilecek potansiyel boşlukları izleme olasılığını artırır. Burada, izlenen ağ trafiği çok büyüktür. Bu nedenle, ağ sensörleri, STS'nin daha iyi verimliliği ve etkinliği ile sonuçlanabilecek bu kadar büyük miktarda verinin üstesinden gelmek için kullanılabilir. Ağ tabanlı STS, ağda bulunan birden çok ana bilgisayarın denetim izlerini analiz eder ve denetler. Bir ağda, izinsiz girişlere yol açabilecek birden fazla olay olabilir ve bu nedenle, izinsiz giriş tespiti için her ağ olayının titizlikle incelenmesi gerekir.

3.2. Tespit Yöntemi

Sistemin kötü niyetli faaliyetlerinin ve müdahaleci davranışlarının tespiti, sistemin altyapısına dayalı olarak STS tarafından benimsenen tespit yöntemi ile gerçekleştirilebilir. Tespit yöntemine dayalı olarak STS, anormallik tabanlı STS ve kötüye kullanım tabanlı STS olarak sınıflandırılır (B. Gupta et al., 2016; Vidal et al., 2020).

3.2.1. Anormallik Tabanlı STS

Anormallik tabanlı saldırı tespitinde, normal kalıplardan herhangi bir sapmayı yansıtmak için önemli kalıplar incelenir. Ağ kalıpları statik ve dinamik olarak analiz edilebilir. Sistemin durumu uzun süre değişmezse, statik olarak kabul edilir. Ağ kalıpları, sistemin yazılım ve donanım kısmı kullanılarak analiz edilebilir. Herhangi bir sistemin donanım bölümlerinin konfigürasyonu statik kalır ve bu nedenle analiz görevini yazılım bölümüne yönlendirir. Sistemin ana görevi, ağ verilerinin durağan kısmına, yani koda dayanır. Örneğin, işletim sistemlerinde, veriler hiçbir zaman kritik yazılımdan önyüklemeye değişmez. Statik anomali tespiti, sistemin bütünlüğünü korumaya odaklanır. Bir hata meydana gelirse veya sistemin bir kısmı bir davetsiz misafir tarafından kurcalanırsa, sistemin statik bir kısmı önceki durumdan sapar.

Dinamik anomali tespitinde denetim izleri ve izlenen ağ trafiği dikkate alınır. Bir sistemin işletim sistemindeki denetim izleri, olay sistemi günlüklerini sıralı bir şekilde yakalar. Dağıtılmış bir ortam olması durumunda, sistem günlüğü olaylarının kısmi sıralaması algılama için yeterlidir. Öte yandan, belirli bir kaynağın kullanım zaman aralığı gibi senaryolar dikkate alınır. Bu gibi durumlarda, normal kaynak tüketimi, eşiklerin tanımlanmasıyla anormal tüketimlerden ayırt edilir. Burada anormalliğin tespiti, bilgisayar kullanıcılarının davranışlarının izlenmesi ve izlenmesi ile sağlanır. Veri veya davranış modeli, gerçek ağ trafiği modellerinden saparsa bir uyarı oluşturulur.

Anormallik tabanlı STS kullanmanın en büyük avantajı, normal trafik modelinden küçük farklılıklar anomali olarak kabul edildiğinden, sıfır gün saldırılarının kalıpları analiz ederek kolayca tanımlanabilmesidir. Ayrıca, hedef işletim ortamlarına bağlı değildir. Bu tür STS'nin dezavantajı, çok sayıda yanlış pozitif üretebilmesidir. Ağ trafiğindeki her anormal kalıbın anormal olması gerekli değildir, güvenlik uzmanı bu yanlış pozitiflerden bazılarını görmezden gelebilir ve bu da gerçek anormal aktivitelerin göz ardı edilmesine neden olabilir. Profili oluştururken ve eğitim aşamasını oluştururken, ağ düzgün izlenmezse bazı kullanıcı eylemlerinin atlanma olasılığı yüksektir. Yanlış alarm oranlarını azaltmak için normal profilin tüm modellerini içeren günlüğün güncellenmesi gerekir.

3.2.2. Kötüye Kullanıma Dayalı STS

Kötüye kullanıma dayalı STS, imza tabanlı STS olarak da adlandırılır. Burada, sistem açıklarına ve önceden bilinen saldırı imzalarına dayalı olarak bir STS oluşturulur. Güvenlik açıklarından yararlanarak sistemi engellemeye çalışan davetsiz misafirleri tanımakla ilgilenir. Sistemin güvenliğini sağlamak için tüm boşluklar ortadan kaldırılmalıdır. İzinsiz giriş tespiti, herhangi bir anormal etkinlik için önleyici bir önlem almak için bir uyarı oluşturulmasıyla sonuçlanan bir dizi adımdır. Kötüye kullanıma dayalı yöntemler, herhangi bir izinsiz giriş etkinliğinin davranışını nasıl farklılaştırdıkları veya şekillendirdikleri açısından farklılık gösterir. İdeal olarak, kötüye kullanıma dayalı algılama sistemi, sistem içindeki olağandışı eylemleri en iyi tanımlayan olayları açıklamak için kuralları kullanır. Farklı izinsiz giriş senaryolarını tahmin etmek için birçok kural formüle edilebilir ve birleştirilebilir. Yanlış kullanıma

dayalı STS, kurallarla eşleşen olayları arar. Olaylar, denetim kayıtları ile daha sonra inceleme için kullanılabilir ve sistem çağrıları incelenerek izlenebilir.

Kötüye kullanım tabanlı STS için zorlu görev, saldırı imzalarını içeren veri tabanını güncel tutmaktır. Yanlış kullanıma dayalı STS, halihazırda mevcut olan saldırı imzası ve yeni saldırılarla bir ilişki kuramadığı için yeni saldırıları tanımakta iyi değildir. Kötüye kullanıma dayalı STS'nin bakımı, sürekli düzeltme ekinin yanı sıra güvenlik açıklarının ve açıklardan yararlanmaların analizini içeren zaman alıcı bir süreçtir. Bir işletim ortamındaki herhangi bir saldırı hakkında bilgi edinmek, işletim sistemi sürümüne, platforma ve uygulamalara bağlıdır. İçeriden saldırıların tespiti daha da zordur. Örneğin, meşru kullanıcı ayrıcalıklarının kötüye kullanılması, sistem tarafından kötü niyetli etkinlik olarak izlenemez veya algılanamaz.

İzinsiz girişin doğru bir şekilde belirlenmesi ile ilgili olarak, kullanıcı uygulamaları ve ağ ortamı ayrılmaz bir rol oynamaktadır. İzinsiz girişleri tespit etmek için hangi tekniklerin uygulanması gerektiği sonucuna varmaya yardımcı olan performans ölçütleri vardır. Bu performans ölçüleri, bir izinsiz girişi doğru şekilde tahmin etme yeteneğinden türetilir. İzinsiz girişi tahmin etmenin performans ölçüleri şu şekilde sınıflandırılır:

- **İzinsiz Giriş, Kötü Amaçlı Olmayan:** Bu, kötü amaçlı olan ancak sistemin izinsiz girişlerin varlığını tespit edemediği etkinlik olarak tanımlanabilir. Bu aynı zamanda Yanlış Negatif (False Negative, FN) olarak da adlandırılabilir.
- **İzinsiz Girişim Yok, Kötü Amaçlı:** İzinsiz giriş içermemesine rağmen kötü niyetli olarak kabul edilen etkinlik olarak tanımlanabilir. Bu aynı zamanda Yanlış Pozitif (False Positive, FP) olarak da adlandırılabilir.
- **İzinsiz Giriş Yok, Kötü Amaçlı Olmayan:** Bu, kötü amaçlı olmayan ve izinsiz giriş yapmayan olarak tanımlanan etkinlik olarak tanımlanabilir. Bu aynı zamanda Gerçek Negatif (True Negative, TN) olarak da adlandırılabilir.
- **İzinsiz Giriş, Kötü Amaçlı:** Bu, araya giren ve doğru bir şekilde kötü niyetli olarak tanımlanan etkinlik olarak tanımlanabilir. Bu, Gerçek Pozitif (True Positive, TP) olarak adlandırılabilir.

3.3. Yanıt Mekanizması

Yanıt mekanizması, bir izinsiz giriş gerçekleştiğinde bir STS'nin yanıt verme şeklidir; aktif veya pasif bir yanıt olabilir (Anwar et al., 2017). Aktif STS müdahale mekanizması, izinsiz girişlerin veya saldırıların tespit edildiği anda güvenlik uzmanına bile bakılmadan anında engellenmesi için oluşturulmuş sistem olarak ifade edilebilir. Gerçek zamanlı olarak meydana gelen saldırıları tespit etme ve yönetme avantajına sahiptir. Aktif yanıt mekanizmaları tarafından kaydedilen yanıtlardan bazıları şunlardır:

- İzinsiz giriş tespit raporu oluşturmak
- Uyarıyı/alarmı tetiklemek
- Ağda meydana gelen olaylar için ekstra bir kayıt olanağına sahip olmak
- Uzaktan meydana gelen olaylar için uzaktan kayıt imkanına sahip olmak
- Şüphelenilen saldırıları anında önlemek için Saldırı Önleme Sistemi kurmak
- Günlüğe kaydedilen etkinliklerin yedeğini almak

Pasif STS yanıt mekanizması, olağandışı bir model veya ağ etkinliğine sahip ağ işlemlerini zorlayarak ağ trafiğini izlemek için oluşturulmuş sistem olarak ifade edilebilir. Ağda meydana gelen izinsiz girişleri proaktif olarak ele alamaz. Pasif yanıt mekanizması olarak kaydedilen yanıtlardan bazıları şunlardır:

- Kullanıcı hesaplarını aniden kilitlemek
- Sistemde çalışan işlemleri askıya almak
- Kullanıcı oturum açma işlemini sonlandırmak ve sistemi kapatmak
- Kullanıcıların IP adreslerinin bloke edilmesi ve port servislerinin kapatılması
- Geçici gölge dosyaları oluşturmak ve kullanma
- Uzaktan oturum açma yoluyla yetkisiz erişimi zorlamak,
- Davetsiz misafiri korkutmak

3.4. Mimari Model

STS için etkin bir kaynak ve veri kombinasyonuna sahip olma gereksinimlerini karşılamak için çeşitli altyapı planları önerilmiştir. Bu altyapılar merkezi ve dağıtık olarak ikiye ayrılabilir (Snapp et al., 1991). Merkezileştirilmiş STS'de, merkezi bir düğüm ağ trafiğini analiz eder ve herhangi bir olağandışı davranış bulunduğunda bir uyarıyı tetikler. Bilgi, diğer ağ düğümlerinden toplanır, burada her düğüm ağ trafiğini

izler ve bilgileri merkezi düğüme gönderir. Bundan sonra, merkezi düğüm, ağa eklenen düğümlerden alınan bilgilere dayanarak uyarılar üretir. Bu tür sistemlerin tek bir temas noktasına sahip olmaları gibi eksiklikleri vardır, bu nedenle merkezi düğüm ele geçirilirse, tüm sistemin savunmasız kalmasına neden olabilir. Merkezi düğüm tarafından işlenen veri ve/veya istek miktarı sınırlı olduğu için işleme ek yüküne yol açar.

Öte yandan, dağıtılmış STS'de, her birim, oluşturulan saldırıyı tespit etme ve yanıt verme yeteneğine sahiptir. Dağıtılmış bir STS, ağaç benzeri bir yapı sergiler. Bunun nedeni, ağ trafiğinin analizi için kullanılan düğümlerin, her birimin birbiriyle aşağıdan yukarıya iletişim kurduğu hiyerarşik bir şekilde yerleştirilmesidir. Bununla birlikte, birimler dağıtıldıkça hata toleransı, yük dengeleme ve içeriden tehdit algılama zorluklarını ortaya çıkarırlar (Snapp et al., 1991).

3.5. Karar Verme

Daha önce tartışıldığı gibi, bir STS yapısı merkezileştirilebilir veya dağıtılabilir. Buna dayanarak, bir STS'nin karar verme şeması, işbirlikçi veya bağımsız olarak gruplandırılabilir (Inayat et al., 2016). Dağıtılmış bir STS'de, çoklu düğümler ağda farklı seviyelerde dağılmıştır. Bu nedenle, analiz edilen faaliyetin müdahaleci olup olmadığına işbirlikçi bir şekilde karar verilir. Karar, istatistiksel teknikler kullanılarak verilirken, merkezi STS'de tek bir düğüm, düğüm tarafından toplanan bilgileri kullanarak kararı bağımsız olarak alır.

Ayrıca, dağıtılmış STS'de, birimler farklı seviyelerde dağıtılabilir veya birimler, bir kümedeki düğümler gibi farklı yerlere dağılır, ancak her düğüm toplu olarak farklı yeteneklere katkıda bulunur. Merkezileştirilmiş STS'deyken, merkezi düğüm tüm ağdan toplanan verileri işler. Tablo 3.2, avantajları ve dezavantajlarıyla birlikte STS'nin özelliklerini özetlemektedir.

Tablo 3.2: STS Özelliklerinin Özeti.

STS'nin Özellikleri	Kategoriler	Avantajlar	Dezavantajlar
Veri Kaynağı	Ana bilgisayar tabanlı STS	Sistem düzeyinde koruma sağlar İçeriden saldırılarla iyi çalışır	Sıfır gün saldırılarında iyi değil Sistem tarafından oluşturulan günlüklere bağlı
	Ağ tabanlı STS	Dış saldırılara karşı güçlü caydırıcılık Ağ düzeyinde güvenlik sağlar	Yanlış pozitiflere açık Sistemle ilgili herhangi bir bilgiye sahip değil
Tespit Yöntemi	Anormali tabanlı STS	Önceden tanımlanmış imzalara bağlı değildir Bilinmeyen saldırıların algılanmasında iyidir	Tasarım hakkında derinlemesine bilgi gerektirir Şifreleme konusunda iyi değil
	Yanlış kullanıma dayalı STS	Orijinaldir ve çok düşük yanlış alarm oranı sergiler İzinsiz girişi azaltmak için çok kesin adımlar sağlar	Sıfır günlerini ve bilinen saldırıların çeşitlerini tespit edemiyor İmza veri tabanında düzenli güncelleme gerekiyor
Yanıt Mekanizması	Aktif	Gerçek zamanlı algılama Proaktif	Paketin derinlemesine analizinden yoksundur
	Pasif	Paketleri analiz etme ve izleme konusunda iyi Operatörü güvenlik açıkları hakkında gösterir	Günlüğe kaydetme etkinliğine dayalıdır Gerçek zamanlı uyarı oluşturamaz Kendi başına işlem yapamaz
Mimari Model	Merkezi	Veriler tek bir noktada saklanır Hesaplama yükü yok	Tek hata noktası Yavaş yanıt süresi
	Dağıtılmış	Eşler Arası iletişim Veriler dağıtılmış şekilde işlenir	Düşük algılama doğruluğu Hata toleransı Yük dengelemede karmaşıklık
Karar verme	İşbirlikçi	Tüm birimlerin ortak kararı	Hesaplama yükü
	Bağımsız	Daha az hesaplama yükü	Karar bireysel düğüme bağlıdır

Anwar ve diğerleri (2017), Gupta ve diğerleri (2016), Inayat ve diğerleri (2016), Snapp ve diğerleri (1991), Vidal ve diğerleri (2020)'den uyarlanmıştır.

3.6. Sınırlamalar ve Kısıtlar

Bir STS, saldırıları nitelemek ve ölçmek için verileri denetlemek için ağ paketlerinin içeriğini incelemek üzere ayarlanabilse de aşağıda verilen bazı eksiklikleri vardır.

- STS, kalıpları tanımlayarak veya veri tabanından saldırı imzasını eşleştirerek henüz tespit edilen saldırıyı veya engelleyemez. İzinsiz girişleri önlemek veya engellemek için STS'nin İzinsiz Giriş Önleme Sistemleri gibi diğer güvenlik mekanizmalarıyla entegre edilmesi gerekir.
- Bir STS, ağın ayrıntılı bir analizini yapar ve ağ etkinliğini izler, ancak bir saldırının algılanması sırasında gerekli eylemi gerçekleştirme yeteneğine sahip değildir. Bu nedenle ağda tespit edilen tehditlere karşı önlem almak için sürekli olarak bir güvenlik görevlisine veya yöneticiye ihtiyaç duyar.
- Bir STS, şifreli ağ paketlerinin işlenmesinde yetersizdir. Şifrelenmiş ağ paketlerini incelemek için ağ araçları gerektirir. Bu, izinsiz giriş tespit edilene kadar sistem kaynaklarını savunmasız bir durumda bırakabilir.
- STS tarafından üretilen yanlış pozitiflerin sayısının yüksek olması, sistemin verimliliğini etkiler.
- Saldırı imzası veri tabanının, yeni saldırı imzalarını içermesi için düzenli olarak güncellenmesi gerekir.
- STS, protokol tabanlı saldırılara açıktır.

4. MATERYAL VE METOT

Bu bölümde çalışmada yer alan materyal ve metotlar açıklanmıştır. Çalışmada kullanılan veri setleri, özellik seçimi ve makine öğrenmesi algoritmaları anlatılmıştır.

4.1. Veri Setleri

Herhangi bir STS'nin performansını ölçmek için; farklı sınıflandırıcıların karşılaştırmasını doğrulayabilen standart bir veri setine yüksek bir gereksinim vardır. Örneğin, 1998'de MIT Lincoln laboratuvarı, DARPA tarafından finanse edilen projeler kapsamında DARPA-98 veri setini geliştirdi. Bu veri seti, son yirmi yılda STS modellerinin performansını değerlendirmek için kullanılmıştır. Analize dayalı olarak, mükerrer kayıtların varlığı, eğitim ve test veri setlerinin kayıtlarında bir dengesizlik ve sentetik trafiğin dikkate alınması gibi birçok dezavantaj tespit edilmiştir (Hugh, 2000).

Bu sınırlamaların üstesinden gelmek için, NSL-KDD gibi veri setinin daha gelişmiş versiyonlarını oluşturmaya yönelik araştırmalar yürütülmektedir (Brown et al., 2009). Literatürde aynı amaç doğrultusunda yola çıkılarak farklı kuruluşlar tarafından farklı veri setleri oluşturulmuş ve yıllar içerisinde geliştirilmeye devam edilmiştir. Çalışmada kullanılan veri setleri ve literatürde bulunan diğer veri setleri aşağıda açıklanmıştır.

4.1.1. Kullanılan Veri Setleri

KDD CUP 99

1999'dan bu yana, KDD CUP 99 anormallik tespit yöntemlerinin değerlendirilmesinde en çok kullanılan veri seti olmuştur (KDD Cup 1999 Data, 2024). Bu veri seti Stolfo ve diğerleri (2000) tarafından, DARPA-98 STS değerlendirme programında elde edilen verilere dayanılarak oluşturulmuştur (Stolfo et al., 2000). KDD veri setinin her bir kaydı 41 adet özellikten oluşmaktadır. Bu kayıtlar normal veya saldırı olarak etiketlenen örneklerden oluşur. Simüle edilen saldırılar; DoS, U2R, R2L ve Probe olarak isimlendirilen kategorilere dahil edilir. Deney aşamasında kullanılan eğitim ve test veri setlerinin içerik örneği aşağıda bulunan Tablo 4.1 ve Tablo 4.2 de verilmiştir.

Tablo 4.1: KDD Cup 99 Eğitim Veri Seti Örnek Değerleri.

Özellikler	Değerler					
"duration"	*"0"*	*"25"*	*"0"*	*"0"*	*"6"*	*"0"*
"protocol_type"	*"tcp"*	*"tcp"*	*"tcp"*	*"icmp"*	*"tcp"*	*"tcp"*
"service"	*"http"*	*"telnet"*	*"telnet"*	*"ecr_i"*	*"http"*	*"finger"*
"flag"	*"SF"*	*"SF"*	*"S0"*	*"SF"*	*"SF"*	*"S0"*
"src_bytes"	*"212"*	*"269"*	*"0"*	*"1032"*	*"0"*	*"0"*
"dst_bytes"	*"786"*	*"2333"*	*"0"*	*"0"*	*"0"*	*"0"*
"land"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"1"*
"wrong_fragment"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"urgent"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"hot"	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_failed_logins"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"logged_in"	*"1"*	*"1"*	*"0"*	*"0"*	*"1"*	*"0"*
"num_compromised"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"root_shell"	*"0"*	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*
"su_attempted"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_root"	*"0"*	*"2"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_file_creations"	*"0"*	*"2"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_shells"	*"0"*	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_access_files"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_outbound_cmds"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_host_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_guest_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"count"	*"8"*	*"1"*	*"6"*	*"316"*	*"1"*	*"1"*
"srv_count"	*"8"*	*"1"*	*"5"*	*"316"*	*"1"*	*"1"*
"serror_rate"	*"0.00"*	*"0.00"*	*"0.83"*	*"0.00"*	*"0.00"*	*"1.00"*
"srv_serror_rate"	*"0.00"*	*"0.00"*	*"1.00"*	*"0.00"*	*"0.00"*	*"1.00"*
"rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"srv_rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"same_srv_rate"	*"1.00"*	*"1.00"*	*"0.83"*	*"1.00"*	*"1.00"*	*"1.00"*
"diff_srv_rate"	*"0.00"*	*"0.00"*	*"0.33"*	*"0.00"*	*"0.00"*	*"0.00"*
"srv_diff_host_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"dst_host_count"	*"8"*	*"69"*	*"5"*	*"148"*	*"61"*	*"1"*
"dst_host_srv_count"	*"99"*	*"2"*	*"6"*	*"3"*	*"2"*	*"8"*
"dst_host_same_srv_rate"	*"1.00"*	*"0.03"*	*"1.00"*	*"0.02"*	*"0.02"*	*"1.00"*
"dst_host_diff_srv_rate"	*"0.00"*	*"0.06"*	*"0.00"*	*"0.02"*	*"1.00"*	*"0.00"*
"dst_host_same_src_port_rate"	*"0.12"*	*"0.01"*	*"0.20"*	*"0.02"*	*"0.02"*	*"1.00"*
"dst_host_srv_diff_host_rate"	*"0.05"*	*"0.00"*	*"0.33"*	*"0.00"*	*"1.00"*	*"0.38"*
"dst_host_serror_rate"	*"0.00"*	*"0.00"*	*"1.00"*	*"0.00"*	*"0.00"*	*"1.00"*
"dst_host_srv_serror_rate"	*"0.00"*	*"0.00"*	*"0.83"*	*"0.00"*	*"0.00"*	*"0.12"*
"dst_host_rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.89"*	*"0.00"*
"dst_host_srv_rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.50"*	*"0.00"*
"attack_cat"	*"normal"*	*"perl"*	*"neptune"*	*"smurf"*	*"ipsweep"*	*"land"*
"label"	*"0"*	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*

Tablo 4.2: KDD Cup 99 Test Veri Seti Örnek Değerleri.

Özellikler	Değerler				
"duration"	*"26"*	*"0"*	*"184"*	*"79"*	*"23"*
"protocol_type"	*"tcp"*	*"tcp"*	*"tcp"*	*"tcp"*	*"tcp"*
"service"	*"ftp"*	*"http"*	*"telnet"*	*"telnet"*	*"telnet"*
"flag"	*"SF"*	*"SF"*	*"SF"*	*"SF"*	*"SF"*
"src_bytes"	*"116"*	*"54540"*	*"1511"*	*"281"*	*"104"*
"dst_bytes"	*"451"*	*"8314"*	*"2957"*	*"1301"*	*"276"*
"land"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"wrong_fragment"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"urgent"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"hot"	*"2"*	*"2"*	*"3"*	*"2"*	*"0"*
"num_failed_logins"	*"0"*	*"0"*	*"0"*	*"0"*	*"5"*
"logged_in"	*"1"*	*"1"*	*"1"*	*"1"*	*"0"*
"num_compromised"	*"0"*	*"1"*	*"2"*	*"1"*	*"0"*
"root_shell"	*"0"*	*"0"*	*"1"*	*"1"*	*"0"*
"su_attempted"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_root"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_file_creations"	*"1"*	*"0"*	*"1"*	*"4"*	*"0"*
"num_shells"	*"0"*	*"0"*	*"0"*	*"2"*	*"0"*
"num_access_files"	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_outbound_cmds"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_host_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_guest_login"	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*
"count"	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*
"srv_count"	*"1"*	*"2"*	*"1"*	*"1"*	*"1"*
"serror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"srv_serror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"srv_rerror_rate"	*"0.00"*	*"0.50"*	*"0.00"*	*"0.00"*	*"0.00"*
"same_srv_rate"	*"1.00"*	*"1.00"*	*"1.00"*	*"1.00"*	*"1.00"*
"diff_srv_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"srv_diff_host_rate"	*"0.00"*	*"1.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"dst_host_count"	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*
"dst_host_srv_count"	*"1"*	*"1"*	*"3"*	*"10"*	*"2"*
"dst_host_same_srv_rate"	*"1.00"*	*"1.00"*	*"1.00"*	*"1.00"*	*"1.00"*
"dst_host_diff_srv_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"dst_host_same_src_port_rate"	*"1.00"*	*"1.00"*	*"1.00"*	*"1.00"*	*"1.00"*
"dst_host_srv_diff_host_rate"	*"0.00"*	*"0.00"*	*"0.67"*	*"0.30"*	*"1.00"*
"dst_host_serror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"dst_host_srv_serror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"dst_host_rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*	*"0.00"*
"dst_host_srv_rerror_rate"	*"0.00"*	*"0.00"*	*"0.00"*	*"0.10"*	*"0.00"*
"attack_cat"	*"ftp_write"*	*"back"*	*"buffer_oflw"*	*"loadmdl"*	*"guess_pwd"*
"label"	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*

Bu çalışmada KDD CUP 99 veri seti üzerindeki deney çalışmaları VS1 ile ifade edilmektedir.

NSL-KDD

KDD Cup 99 veri setinde üzerinde yapılan çalışmalar, sistemlerin performansını oldukça etkileyen önemli sorunları barındırdığı ve anormallik tespit yaklaşımlarının çok zayıf tahmin edilmesine yol açtığını göstermiştir. Bu sorunları çözmek için, KDD veri seti içerisinden seçilen kayıtlar ile NSL-KDD isimli yeni bir veri seti önerilmiştir (NSL-KDD Datasets Research, 2024). NSL-KDD veri kümesinin avantajı; eğitim ve test veri setinde gereksiz, dolayısı ile mükerrer kayıt yoktur. Bundan dolayı sınıflandırma sonucunda yanlış pozitif sayısı düşüktür.

Eğitim veri seti, test veri setinde mevcut olan 37 saldırı türünden farklı 21 saldırı türünden oluşur. Bilinen saldırı türleri, eğitim veri setinde mevcut olanlardır; yeni saldırılar ise test veri setindeki ek saldırılardır, yani eğitim veri setlerinde mevcut değildir. Saldırı türleri dört kategoriye ayrılır; DoS, U2R, R2L ve Probe (NSL-KDD Datasets Research, 2024). Deney aşamasında kullanılan eğitim ve test veri setlerinin içerik örneği aşağıda bulunan Tablo 4.3 ve Tablo 4.4 de verilmiştir.

Tablo 4.3: NSL KDD Eğitim Veri Seti Örnek Değerleri.

Özellikler	Değerler					
"duration"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"proto"	*"tcp"*	*"tcp"*	*"icmp"*	*"tcp"*	*"icmp"*	*"tcp"*
"service"	*"ftp_data"*	*"name"*	*"eco_i"*	*"private"*	*"eco_i"*	*"private"*
"flag"	*"SF"*	*"S0"*	*"SF"*	*"REJ"*	*"SF"*	*"REJ"*
"src_bytes"	*"491"*	*"0"*	*"8"*	*"0"*	*"8"*	*"0"*
"dst_bytes"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"land"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"wrong_fragment"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"urgent"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"hot"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_failed_logins"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"logged_in"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_compromised"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"root_shell"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"su_attempted"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_root"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_file_creations"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_shells"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_access_files"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_outbound_cmds"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_host_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_guest_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"count"	*"2"*	*"233"*	*"1"*	*"2"*	*"1"*	*"175"*
"srv_count"	*"2"*	*"1"*	*"12"*	*"1"*	*"15"*	*"1"*
"serror_rate"	*"0"*	*"1"*	*"0"*	*"0"*	*"0"*	*"0.1"*
"srv_serror_rate"	*"0"*	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*
"rerror_rate"	*"0"*	*"0"*	*"0"*	*"1"*	*"0"*	*"0.89"*
"srv_rerror_rate"	*"0"*	*"0"*	*"0"*	*"1"*	*"0"*	*"1"*
"same_srv_rate"	*"1"*	*"0"*	*"1"*	*"0.5"*	*"1"*	*"0.01"*
"diff_srv_rate"	*"0"*	*"0.06"*	*"0"*	*"1"*	*"0"*	*"1"*
"srv_diff_host_rate"	*"0"*	*"0"*	*"1"*	*"0"*	*"1"*	*"0"*
"dst_host_count"	*"150"*	*"255"*	*"1"*	*"255"*	*"2"*	*"255"*
"dst_host_srv_count"	*"25"*	*"1"*	*"53"*	*"1"*	*"46"*	*"1"*
"dst_host_same_srv_rate"	*"0.17"*	*"0"*	*"1"*	*"0"*	*"1"*	*"0"*
"dst_host_diff_srv_rate"	*"0.03"*	*"0.07"*	*"0"*	*"0.31"*	*"0"*	*"0.84"*
"dst_host_same_src_port_rate"	*"0.17"*	*"0"*	*"1"*	*"0.28"*	*"1"*	*"0"*
"dst_host_srv_diff_host_rate"	*"0"*	*"0"*	*"0.51"*	*"0"*	*"0.26"*	*"0"*
"dst_host_serror_rate"	*"0"*	*"1"*	*"0"*	*"0"*	*"0"*	*"0.07"*
"dst_host_srv_serror_rate"	*"0"*	*"1"*	*"0"*	*"0"*	*"0"*	*"0"*
"dst_host_rerror_rate"	*"0.05"*	*"0"*	*"0"*	*"0.29"*	*"0"*	*"0.62"*
"dst_host_srv_rerror_rate"	*"0"*	*"0"*	*"0"*	*"1"*	*"0"*	*"1"*
"attack_cat"	*"normal"*	*"neptune"*	*"ipswp"*	*"portswp"*	*"nmap"*	*"satan"*
"label"	*"0"*	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*

Tablo 4.4: NSL KDD Test Veri Seti Örnek Değerleri.

Özellikler	Değerler					
"duration"	*"2"*	*"0"*	*"1"*	*"0"*	*"0"*	*"2066"*
"proto"	*"tcp"*	*"icmp"*	*"tcp"*	*"tcp"*	*"tcp"*	*"tcp"*
"service"	*"ftp_data"*	*"eco_i"*	*"telnet"*	*"telnet"*	*"other"*	*"http"*
"flag"	*"SF"*	*"SF"*	*"RSTO"*	*"SF"*	*"REJ"*	*"RSTR"*
"src_bytes"	*"12983"*	*"20"*	*"0"*	*"129"*	*"0"*	*"56504"*
"dst_bytes"	*"0"*	*"0"*	*"15"*	*"174"*	*"0"*	*"0"*
"land"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"wrong_fragment"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"urgent"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"hot"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_failed_logins"	*"0"*	*"0"*	*"0"*	*"1"*	*"0"*	*"0"*
"logged_in"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"1"*
"num_compromised"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"root_shell"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"su_attempted"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_root"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_file_creations"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_shells"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_access_files"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"num_outbound_cmds"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_host_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"is_guest_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"count"	*"1"*	*"1"*	*"1"*	*"1"*	*"2"*	*"8"*
"srv_count"	*"1"*	*"65"*	*"8"*	*"1"*	*"1"*	*"8"*
"serror_rate"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"srv_serror_rate"	*"0"*	*"0"*	*"0.12"*	*"0"*	*"0"*	*"0"*
"rerror_rate"	*"0"*	*"0"*	*"1"*	*"0"*	*"0.5"*	*"1"*
"srv_rerror_rate"	*"0"*	*"0"*	*"0.5"*	*"0"*	*"1"*	*"1"*
"same_srv_rate"	*"1"*	*"1"*	*"1"*	*"1"*	*"0.5"*	*"1"*
"diff_srv_rate"	*"0"*	*"0"*	*"0"*	*"0"*	*"1"*	*"0"*
"srv_diff_host_rate"	*"0"*	*"1"*	*"0.75"*	*"0"*	*"0"*	*"0"*
"dst_host_count"	*"134"*	*"3"*	*"29"*	*"255"*	*"255"*	*"255"*
"dst_host_srv_count"	*"86"*	*"57"*	*"86"*	*"255"*	*"2"*	*"244"*
"dst_host_same_srv_rate"	*"0.61"*	*"1"*	*"0.31"*	*"1"*	*"0.01"*	*"0.96"*
"dst_host_diff_srv_rate"	*"0.04"*	*"0"*	*"0.17"*	*"0"*	*"0.02"*	*"0.01"*
"dst_host_same_src_port_rate"	*"0.61"*	*"1"*	*"0.03"*	*"0"*	*"0"*	*"0"*
"dst_host_srv_diff_host_rate"	*"0.02"*	*"0.28"*	*"0.02"*	*"0"*	*"0"*	*"0"*
"dst_host_serror_rate"	*"0"*	*"0"*	*"0"*	*"0.01"*	*"0"*	*"0.02"*
"dst_host_srv_serror_rate"	*"0"*	*"0"*	*"0"*	*"0.01"*	*"0"*	*"0.02"*
"dst_host_rerror_rate"	*"0"*	*"0"*	*"0.83"*	*"0.02"*	*"0.01"*	*"0.51"*
"dst_host_srv_rerror_rate"	*"0"*	*"0"*	*"0.71"*	*"0.02"*	*"1"*	*"0.53"*
"attack_cat"	*"normal"*	*"saint"*	*"mscan"*	*"guess_pwd"*	*"httptnl"*	*"apch2"*
"label"	*"0"*	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*

Bu çalışmada NSL KDD veri seti üzerindeki deney çalışmaları VS2 ile ifade edilmektedir.

UNSW-NB15

Ağ saldırı tespit sistemlerinin başarısı, normal ve anormal davranışları içeren kapsamlı bir veri seti gerektiren saldırıları tanımlama performansına göre değerlendirilir (Gogoi Prasantaand Bhuyan, 2012). KDDCUP 99 ve NSL-KDD veri setleri, ağ tabanlı STS performansını değerlendirmek için yaygın olarak benimsenen ve kullanılan veri setleridir. KDDCUP 99 veri setinin çok fazla sayıda gereksiz kayıt içermesi ve NSL-KDD veri setinin güncel saldırı ortamlarını simüle edememesi bu veri setlerinin eksik yönleridir.

Yukarıdaki nedenler, Avustralya Siber Güvenlik Merkezi'ndeki (ACCS) 2 siber güvenlik araştırma grubu ve dünya genelinde bu alanda çalışan diğer araştırmacılar için ciddi bir mücadeleye yol açmıştır. Araştırma gurubu tarafından gerçek modern normal davranışlar ile sentetik saldırı faaliyetlerinin bir melezi olan UNSW-NB15 veri seti oluşturulmuştur. Veri seti 49 farklı özellikten oluşmaktadır (UNSW-NB15 Dataset, 2024). Aşağıdaki tabloda eğitim ve test veri kümelerinden örneklere yer verilmiştir. Deney aşamasında kullanılan eğitim ve test veri setlerinin içerik örneği aşağıda bulunan Tablo 4.5 ve Tablo 4.6 da verilmiştir.

Tablo 4.5: UNSW-NB15 Eğitim Veri Seti Örnek Değerleri.

Özellikler	Değerler			
"dur"	*"0.121478"*	*"49497181"*	*"0.682342"*	*"0.000009"*
"proto"	*"tcpXX"*	*"tcp"*	*"tcp"*	*"udp"*
"service"	*"-"*	*"-"*	*"-"*	*"-"*
"state"	*"FIN"*	*"FIN"*	*"FIN"*	*"INT"*
"spkts"	*"6"*	*"28"*	*"10"*	*"2"*
"dpkts"	*"4"*	*"24"*	*"8"*	*"0"*
"sbytes"	*"258"*	*"1726"*	*"1248"*	*"256"*
"dbytes"	*"172"*	*"1356"*	*"354"*	*"0"*
"rate"	*"7408749"*	*"1030362"*	*"24914193"*	*"1111111072"*
"sttl"	*"252"*	*"254"*	*"254"*	*"254"*
"dttl"	*"254"*	*"252"*	*"252"*	*"0"*
"sload"	*"1415894238"*	*"269106232"*	*"1317814258"*	*"113777776"*
"dload"	*"8495365234"*	*"210112976"*	*"3634541016"*	*"0"*
"sloss"	*"0"*	*"16"*	*"2"*	*"0"*
"dloss"	*"0"*	*"10"*	*"1"*	*"0"*
"sinpkt"	*"242956"*	*"183322874"*	*"69923222"*	*"0.009"*
"dinpkt"	*"8375"*	*"2149881"*	*"86229"*	*"0"*
"sjit"	*"30177547"*	*"2928116292"*	*"4765382053"*	*"0"*
"djit"	*"11830604"*	*"9819529"*	*"118231945"*	*"0"*
"swin"	*"255"*	*"255"*	*"255"*	*"0"*
"stcpb"	*"621772692"*	*"3932843103"*	*"596039702"*	*"0"*
"dtcpb"	*"2202533631"*	*"750857658"*	*"3669691190"*	*"0"*
"dwin"	*"255"*	*"255"*	*"255"*	*"0"*
"tcprtt"	*"0"*	*"0.092986"*	*"0.115232"*	*"0"*
"synack"	*"0"*	*"0.031755"*	*"0.078729"*	*"0"*
"ackdat"	*"0"*	*"0.061231"*	*"0.036503"*	*"0"*
"smean"	*"43"*	*"62"*	*"125"*	*"128"*
"dmean"	*"43"*	*"57"*	*"44"*	*"0"*
"trans_depth"	*"0"*	*"0"*	*"1"*	*"0"*
"response_body_len"	*"0"*	*"0"*	*"0"*	*"0"*
"ct_srv_src"	*"1"*	*"11"*	*"1"*	*"2"*
"ct_state_ttl"	*"0"*	*"1"*	*"1"*	*"2"*
"ct_dst_ltm"	*"1"*	*"4"*	*"1"*	*"1"*
"ct_src_dport_ltm"	*"1"*	*"4"*	*"1"*	*"1"*
"ct_dst_sport_ltm"	*"1"*	*"3"*	*"1"*	*"1"*
"ct_dst_src_ltm"	*"1"*	*"11"*	*"1"*	*"1"*
"is_ftp_login"	*"0"*	*"0"*	*"0"*	*"0"*
"ct_ftp_cmd"	*"0"*	*"0"*	*"0"*	*"0"*
"ct_flw_http_mthd"	*"0"*	*"0"*	*"1"*	*"0"*
"ct_src_ltm"	*"1"*	*"4"*	*"1"*	*"2"*
"ct_srv_dst"	*"1"*	*"11"*	*"1"*	*"1"*
"is_sm_ips_ports"	*"0"*	*"0"*	*"0"*	*"0"*
"attack_cat"	*"Normal"*	*"Fuzzers"*	*"Exploits"*	*"Shellcode"*
"label"	*"0"*	*"1"*	*"1"*	*"1"*

Tablo 4.6: UNSW-NB15 Test Veri Seti Örnek Değerleri.

Özellikler	Değerler				
"dur"	*"0.000011"*	*"0.921987"*	*"0.921987"*	*"0.921987"*	*"0.000009"*
"proto"	*"udp"*	*"ospf"*	*"ospf"*	*"ospf"*	*"sctp"*
"service"	*"-"*	*"-"*	*"-"*	*"-"*	*"-"*
"state"	*"INT"*	*"INT"*	*"INT"*	*"INT"*	*"INT"*
"spkts"	*"2"*	*"20"*	*"20"*	*"20"*	*"2"*
"dpkts"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"sbytes"	*"496"*	*"1280"*	*"1280"*	*"1280"*	*"104"*
"dbytes"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"rate"	*"909090902"*	*"20607666"*	*"20607666"*	*"20607666"*	*"1111111072"*
"sttl"	*"254"*	*"254"*	*"254"*	*"254"*	*"254"*
"dttl"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"sload"	*"180363632"*	*"10551125"*	*"10551125"*	*"10551125"*	*"46222220"*
"dload"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"sloss"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"dloss"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"sinpkt"	*"0.011"*	*"48525633"*	*"48525633"*	*"48525633"*	*"0.009"*
"dinpkt"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"sjit"	*"0"*	*"52253805"*	*"52253805"*	*"52253805"*	*"0"*
"djit"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"swin"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"stcpb"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"dtcpb"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"dwin"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"tcprtt"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"synack"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"ackdat"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"smean"	*"248"*	*"64"*	*"64"*	*"64"*	*"52"*
"dmean"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"trans_depth"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"response_body_len"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"ct_srv_src"	*"2"*	*"1"*	*"1"*	*"1"*	*"1"*
"ct_state_ttl"	*"2"*	*"2"*	*"2"*	*"2"*	*"2"*
"ct_dst_ltm"	*"1"*	*"1"*	*"1"*	*"1"*	*"2"*
"ct_src_dport_ltm"	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*
"ct_dst_sport_ltm"	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*
"ct_dst_src_ltm"	*"2"*	*"2"*	*"2"*	*"2"*	*"2"*
"is_ftp_login"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"ct_ftp_cmd"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"ct_flw_http_mthd"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"ct_src_ltm"	*"1"*	*"1"*	*"1"*	*"1"*	*"1"*
"ct_srv_dst"	*"2"*	*"1"*	*"1"*	*"1"*	*"1"*
"is_sm_ips_ports"	*"0"*	*"0"*	*"0"*	*"0"*	*"0"*
"attack_cat"	*"Normal"*	*"Recon"*	*"Backdoor"*	*"DoS"*	*"Exploits"*
"label"	*"0"*	*"1"*	*"1"*	*"1"*	*"1"*

Bu çalışmada UNSW-NB15 veri seti üzerindeki deney çalışmaları VS3 ile ifade edilmektedir.

CSE-CIC-IDS2018

CSE-CIC-IDS2018 gibi veri kümeleri, ağ trafiği için anormallik tabanlı izinsiz giriş tespitine ilişkin tahmin modellerini eğitmek amacıyla oluşturulmuştur (Sharafaldinet al., 2018). CSE-CIC-IDS2018 tamamen yeni bir proje değil ISCXIDS2012 veri kümesinden devam ederek geliştirilen bir veri setidir (Shiravi et al., 2012). ISCX2012 olarak da adlandırılan ISCXIDS2012 veri seti, 2012 yılında New Brunswick Üniversitesi'ndeki (UNB) Bilgi Güvenliği Mükemmeliyet Merkezi (ISCX) tarafından yedi günlük bir süre boyunca oluşturulan hem normal hem de anormal ağ trafiğini içerir.

2017 yılında ISCX2012'nin yaratıcıları ve Kanada Siber Güvenlik Enstitüsü (CIC), veri kümesinin yalnızca altı trafik protokolüyle; Postane Protokolü 3 (POP3), Güvenli Kabuk Protokolü (SSH), Basit Posta Aktarım Protokolü (SMTP), Hiper Metin Transfer Protokolü (HTTP), IMAP, FTP sınırlı olduğu gerçeğinden yola çıkarak harekete geçmiştir. Çünkü ISCX2012 veri seti, gerçek dünyadaki mevcut ağ trafiğinin yaklaşık %70'ini oluşturan önemli bir protokol olan Güvenli Hiper Metin Transfer Protokolü (HTTPS)'nin temsil etmemektedir (Sharafaldin et al., 2018). Buna ek olarak simüle edilen saldırıların dağılımının da gerçeğe uygun olmamasından yola çıkılarak, CIC-IDS2017 bu eksiklikleri gidermek amacıyla yayımlanmıştır. CIC-IDS2017 veri seti, 80 adet özelliğe sahip olması nedeniyle makine öğrenimini kolaylaştırmıştır. Bunun yanısıra, veri kümesinin sınıf dengesizliği bulunmaktadır. Çoğunluk ve azınlık sınıfları arasındaki eşitsiz dağılımdan kaynaklanan bir olgu olan sınıf dengesizliği, büyük veri çalışmasında sonuçları çarpıtabilir. Ayrıntılı düzeyde bakıldığında CICIDS2017, bazı bireysel saldırı türlerine göre yüksek sınıf dengesizliğine sahiptir (Shiravi et al., 2012).

2018 yılında izinsiz giriş tespit veri kümesinin en son yinelemesi CSE-CIC-IDS2018 yayınlanmıştır (IDS 2018 Datasets Research, 2024). Güncellenen versiyonda ayrıca sınıf dengesizliği mevcut ve yapısal olarak CICIDS2017'ye benzemektedir. CICIDS2018 olarak da adlandırılan CSE-CIC-IDS2018 veri seti, fark olarak simüle edilmiş istemci hedefleri ve saldırı makinelerinden oluşan çok daha geniş bir ağdan

hazırlanmıştır. Deney aşamasında kullanılan eğitim ve test veri setlerinin içerik örneği aşağıda bulunan Tablo 4.7 ve Tablo 4.8 de verilmiştir.

Tablo 4.7: CSE-CIC-IDS2018 Eğitim Veri Seti Örnek Değerleri.

Özellikler	Değerler		
"Dst Port"	*"80"*	*"21"*	*"22"*
"Protocol"	*"6"*	*"6"*	*"6"*
"Flow Duration"	*"477161"*	*"2"*	*"6"*
"Tot Fwd Pkts"	*"5"*	*"1"*	*"1"*
"Tot Bwd Pkts"	*"3"*	*"1"*	*"1"*
"TotLen Fwd Pkts"	*"211"*	*"0"*	*"0"*
"TotLen Bwd Pkts"	*"463"*	*"0"*	*"0"*
"Fwd Pkt Len Max"	*"211"*	*"0"*	*"0"*
"Fwd Pkt Len Min"	*"0"*	*"0"*	*"0"*
"Fwd Pkt Len Mean"	*"42.2"*	*"0"*	*"0"*
"Fwd Pkt Len Std"	*"943620686505"*	*"0"*	*"0"*
"Bwd Pkt Len Max"	*"463"*	*"0"*	*"0"*
"Bwd Pkt Len Min"	*"0"*	*"0"*	*"0"*
"Bwd Pkt Len Mean"	*"1543333333333"*	*"0"*	*"0"*
"Bwd Pkt Len Std"	*"2673131746348"*	*"0"*	*"0"*
"Flow Byts/s"	*"14125211406632"*	*"0"*	*"0"*
"Flow Pkts/s"	*"167658295628"*	*"1000000"*	*"333333333333333"*
"Flow IAT Mean"	*"681658571428572"*	*"2"*	*"6"*
"Flow IAT Std"	*"116324530541969"*	*"0"*	*"0"*
"Flow IAT Max"	*"238504"*	*"2"*	*"6"*
"Flow IAT Min"	*"12"*	*"2"*	*"6"*
"Fwd IAT Tot"	*"477161"*	*"0"*	*"0"*
"Fwd IAT Mean"	*"119290.25"*	*"0"*	*"0"*
"Fwd IAT Std"	*"137729560342905"*	*"0"*	*"0"*
"Fwd IAT Max"	*"238719"*	*"0"*	*"0"*
"Fwd IAT Min"	*"12"*	*"0"*	*"0"*
"Bwd IAT Tot"	*"238618"*	*"0"*	*"0"*
"Bwd IAT Mean"	*"119309"*	*"0"*	*"0"*
"Bwd IAT Std"	*"168449805841384"*	*"0"*	*"0"*
"Bwd IAT Max"	*"238421"*	*"0"*	*"0"*
"Bwd IAT Min"	*"197"*	*"0"*	*"0"*
"Fwd PSH Flags"	*"0"*	*"0"*	*"0"*
"Bwd PSH Flags"	*"0"*	*"0"*	*"0"*
"Fwd URG Flags"	*"0"*	*"0"*	*"0"*
"Bwd URG Flags"	*"0"*	*"0"*	*"0"*
"Fwd Header Len"	*"168"*	*"40"*	*"32"*
"Bwd Header Len"	*"104"*	*"20"*	*"32"*

Tablo 4.7'nin devamıdır.

Özellikler	Değerler		
"Fwd Pkts/s"	*"104786434767"*	*"500000"*	*"166666666666667"*
"Bwd Pkts/s"	*"6287186086"*	*"500000"*	*"166666666666667"*
"Pkt Len Min"	*"0"*	*"0"*	*"0"*
"Pkt Len Max"	*"463"*	*"0"*	*"0"*
"Pkt Len Mean"	*"748888888889"*	*"0"*	*"0"*
"Pkt Len Std"	*"1614058893322"*	*"0"*	*"0"*
"Pkt Len Var"	*"260518611111111"*	*"0"*	*"0"*
"FIN Flag Cnt"	*"0"*	*"0"*	*"0"*
"SYN Flag Cnt"	*"0"*	*"0"*	*"0"*
"RST Flag Cnt"	*"0"*	*"0"*	*"0"*
"PSH Flag Cnt"	*"1"*	*"1"*	*"0"*
"ACK Flag Cnt"	*"0"*	*"0"*	*"1"*
"URG Flag Cnt"	*"0"*	*"0"*	*"1"*
"CWE Flag Count"	*"0"*	*"0"*	*"0"*
"ECE Flag Cnt"	*"0"*	*"0"*	*"0"*
"Down/Up Ratio"	*"0"*	*"1"*	*"1"*
"Pkt Size Avg"	*"84.25"*	*"0"*	*"0"*
"Fwd Seg Size Avg"	*"42.2"*	*"0"*	*"0"*
"Bwd Seg Size Avg"	*"1543333333333"*	*"0"*	*"0"*
"Fwd Byts/b Avg"	*"0"*	*"0"*	*"0"*
"Fwd Pkts/b Avg"	*"0"*	*"0"*	*"0"*
"Fwd Blk Rate Avg"	*"0"*	*"0"*	*"0"*
"Bwd Byts/b Avg"	*"0"*	*"0"*	*"0"*
"Bwd Pkts/b Avg"	*"0"*	*"0"*	*"0"*
"Bwd Blk Rate Avg"	*"0"*	*"0"*	*"0"*
"Subflow Fwd Pkts"	*"5"*	*"1"*	*"1"*
"Subflow Fwd Byts"	*"211"*	*"0"*	*"0"*
"Subflow Bwd Pkts"	*"3"*	*"1"*	*"1"*
"Subflow Bwd Byts"	*"463"*	*"0"*	*"0"*
"Init Fwd Win Byts"	*"14480"*	*"26883"*	*"241"*
"Init Bwd Win Byts"	*"219"*	*"0"*	*"230"*
"Fwd Act Data Pkts"	*"1"*	*"0"*	*"0"*
"Fwd Seg Size Min"	*"32"*	*"40"*	*"32"*
"Active Mean"	*"0"*	*"0"*	*"0"*
"Active Std"	*"0"*	*"0"*	*"0"*
"Active Max"	*"0"*	*"0"*	*"0"*
"Active Min"	*"0"*	*"0"*	*"0"*
"Idle Mean"	*"0"*	*"0"*	*"0"*
"Idle Std"	*"0"*	*"0"*	*"0"*
"Idle Max"	*"0"*	*"0"*	*"0"*
"Idle Min"	*"0"*	*"0"*	*"0"*
"Label"	*"Benign"*	*"FTP-BruteForce"*	*"SSH-Bruteforce"*

Tablo 4.8: CSE-CIC-IDS2018 Test Veri Seti Örnek Değerleri.

Özellikler	Değerler			
"Dst Port"	*"80"*	*"80"*	*"443"*	*"8080"*
"Protocol"	*"6"*	*"6"*	*"6"*	*"6"*
"Flow Duration"	*"11673"*	*"61"*	*"223"*	*"465"*
"Tot Fwd Pkts"	*"3"*	*"2"*	*"3"*	*"2"*
"Tot Bwd Pkts"	*"4"*	*"0"*	*"0"*	*"0"*
"TotLen Fwd Pkts"	*"340"*	*"0"*	*"77"*	*"0"*
"TotLen Bwd Pkts"	*"935"*	*"0"*	*"0"*	*"0"*
"Fwd Pkt Len Max"	*"340"*	*"0"*	*"46"*	*"0"*
"Fwd Pkt Len Min"	*"0"*	*"0"*	*"0"*	*"0"*
"Fwd Pkt Len Mean"	*"1133333333"*	*"0"*	*"2566666667"*	*"0"*
"Fwd Pkt Len Std"	*"1962990915"*	*"0"*	*"2345918441"*	*"0"*
"Bwd Pkt Len Max"	*"935"*	*"0"*	*"0"*	*"0"*
"Bwd Pkt Len Min"	*"0"*	*"0"*	*"0"*	*"0"*
"Bwd Pkt Len Mean"	*"233.75"*	*"0"*	*"0"*	*"0"*
"Bwd Pkt Len Std"	*"467.5"*	*"0"*	*"0"*	*"0"*
"Flow Byts/s"	*"1092264199"*	*"0"*	*"3452914798"*	*"0"*
"Flow Pkts/s"	*"5996744624"*	*"327868852459016"*	*"134529148"*	*"4301075269"*
"Flow IAT Mean"	*"1945.5"*	*"61"*	*"111.5"*	*"465"*
"Flow IAT Std"	*"4176539991"*	*"0"*	*"1308147545"*	*"0"*
"Flow IAT Max"	*"10446"*	*"61"*	*"204"*	*"465"*
"Flow IAT Min"	*"6"*	*"61"*	*"19"*	*"465"*
"Fwd IAT Tot"	*"10824"*	*"61"*	*"223"*	*"465"*
"Fwd IAT Mean"	*"5412"*	*"61"*	*"111.5"*	*"465"*
"Fwd IAT Std"	*"7119151073"*	*"0"*	*"1308147545"*	*"0"*
"Fwd IAT Max"	*"10446"*	*"61"*	*"204"*	*"465"*
"Fwd IAT Min"	*"378"*	*"61"*	*"19"*	*"465"*
"Bwd IAT Tot"	*"11665"*	*"0"*	*"0"*	*"0"*
"Bwd IAT Mean"	*"3888333333"*	*"0"*	*"0"*	*"0"*
"Bwd IAT Std"	*"6017873905"*	*"0"*	*"0"*	*"0"*
"Bwd IAT Max"	*"10822"*	*"0"*	*"0"*	*"0"*
"Bwd IAT Min"	*"24"*	*"0"*	*"0"*	*"0"*
"Fwd PSH Flags"	*"0"*	*"0"*	*"1"*	*"0"*
"Bwd PSH Flags"	*"0"*	*"0"*	*"0"*	*"0"*
"Fwd URG Flags"	*"0"*	*"0"*	*"0"*	*"0"*
"Bwd URG Flags"	*"0"*	*"0"*	*"0"*	*"0"*
"Fwd Header Len"	*"72"*	*"40"*	*"60"*	*"40"*
"Bwd Header Len"	*"92"*	*"0"*	*"0"*	*"0"*
"Fwd Pkts/s"	*"257003341"*	*"327868852459016"*	*"134529148"*	*"4301075269"*

Tablo 4.8'in devamıdır.

Özellikler	Değerler			
"Bwd Pkts/s"	*"3426711214"*	*"0"*	*"0"*	*"0"*
"Pkt Len Min"	*"0"*	*"0"*	*"0"*	*"0"*
"Pkt Len Max"	*"935"*	*"0"*	*"46"*	*"0"*
"Pkt Len Mean"	*"159375"*	*"0"*	*"30.75"*	*"0"*
"Pkt Len Std"	*"33522314"*	*"0"*	*"2168524844"*	*"0"*
"Pkt Len Var"	*"1123745536"*	*"0"*	*"470.25"*	*"0"*
"FIN Flag Cnt"	*"0"*	*"0"*	*"0"*	*"0"*
"SYN Flag Cnt"	*"0"*	*"0"*	*"1"*	*"0"*
"RST Flag Cnt"	*"1"*	*"0"*	*"0"*	*"0"*
"PSH Flag Cnt"	*"1"*	*"0"*	*"0"*	*"0"*
"ACK Flag Cnt"	*"0"*	*"1"*	*"1"*	*"1"*
"URG Flag Cnt"	*"0"*	*"0"*	*"0"*	*"0"*
"CWE Flag Count"	*"0"*	*"0"*	*"0"*	*"0"*
"ECE Flag Cnt"	*"1"*	*"0"*	*"0"*	*"0"*
"Down/Up Ratio"	*"1"*	*"0"*	*"0"*	*"0"*
"Pkt Size Avg"	*"1821428571"*	*"0"*	*"41"*	*"0"*
"Fwd Seg Size Avg"	*"1133333333"*	*"0"*	*"2566666667"*	*"0"*
"Bwd Seg Size Avg"	*"233.75"*	*"0"*	*"0"*	*"0"*
"Fwd Byts/b Avg"	*"0"*	*"0"*	*"0"*	*"0"*
"Fwd Pkts/b Avg"	*"0"*	*"0"*	*"0"*	*"0"*
"Fwd Blk Rate Avg"	*"0"*	*"0"*	*"0"*	*"0"*
"Bwd Byts/b Avg"	*"0"*	*"0"*	*"0"*	*"0"*
"Bwd Pkts/b Avg"	*"0"*	*"0"*	*"0"*	*"0"*
"Bwd Blk Rate Avg"	*"0"*	*"0"*	*"0"*	*"0"*
"Subflow Fwd Pkts"	*"3"*	*"2"*	*"3"*	*"2"*
"Subflow Fwd Byts"	*"340"*	*"0"*	*"77"*	*"0"*
"Subflow Bwd Pkts"	*"4"*	*"0"*	*"0"*	*"0"*
"Subflow Bwd Byts"	*"935"*	*"0"*	*"0"*	*"0"*
"Init Fwd Win Byts"	*"65535"*	*"2051"*	*"256"*	*"2052"*
"Init Bwd Win Byts"	*"219"*	*"-1"*	*"-1"*	*"-1"*
"Fwd Act Data Pkts"	*"1"*	*"0"*	*"1"*	*"0"*
"Fwd Seg Size Min"	*"20"*	*"20"*	*"20"*	*"20"*
"Active Mean"	*"0"*	*"0"*	*"0"*	*"0"*
"Active Std"	*"0"*	*"0"*	*"0"*	*"0"*
"Active Max"	*"0"*	*"0"*	*"0"*	*"0"*
"Active Min"	*"0"*	*"0"*	*"0"*	*"0"*
"Idle Mean"	*"0"*	*"0"*	*"0"*	*"0"*
"Idle Std"	*"0"*	*"0"*	*"0"*	*"0"*
"Idle Max"	*"0"*	*"0"*	*"0"*	*"0"*
"Idle Min"	*"0"*	*"0"*	*"0"*	*"0"*
"Label"	*"Benign"*	*"SQL Injection"*	*"Infilteration"*	*"Bot"*

Bu çalışmada CSE-CIC-IDS2018 veri seti üzerindeki deney çalışmaları VS4 ile ifade edilmektedir.

4.1.2. Diğer Veri Setleri

DARPA

Bu veri seti, MIT Lincoln Laboratuvarı'nın DARPA destekli projesinin bir parçasıydı ve saldırı senaryoları ve normal trafik altında ağ trafiği analizi gereksinimlerini karşılamak için geliştirildi. SMTP, FTP, HTTP ve Telnet gibi hizmet kategorilerini içerir. Simüle edilen saldırılar DoS, Probe, R2L ve U2R adı altında kategorize edilir. Bu veri kümesinin gerçek ağ trafiğinden yoksun olması, yakalanan verilerin düzenlilik göstermemesi ve yanlış pozitiflerin olmaması gibi bazı sınırlamaları vardır. Bu veri seti, yeni saldırıları belirlemek ve sınıflandırmak için STS'nin performansını değerlendirme yeteneğine sahip değildir (Brown et al., 2009; Hugh, 2000).

DEFCON

DEFCON veri seti, 2000 yılında Shmoo Group tarafından "bayrak yakalama" yarışması sırasında üretilen trafiği yakalayarak oluşturulmuştur (Migliavacca et al., 2010). Port tarama, arabellek taşması, süpürme, yetkisiz erişim, telnet ve FTP protokolü saldırıları ve bozuk paketler gibi saldırılardan oluşur. Bu veri kümesi, gerçek ağ trafiğinden gerçek dışı olma sınırlamasına sahiptir. Birçok uyarı korelasyon yöntemini uygulamak için kullanılabilir (Migliavacca et al., 2010).

CAIDA

Uygulamalı İnternet Veri Analizi Merkezi (CAIDA), ağ paketinin kaynağı ve hedefiyle ilgili olarak sağladıkları yük bilgilerine ve saldırı türlerine çok özel bir veri kümesi oluşturmuştur. İlk olarak 2002 yılında San Jose'de OC48 bağlantısı kullanılarak veriler analiz edilerek inşa edilmiştir. Daha sonra 2016 yılında Paket Yakalama (PCAP) dosyalarından yakalanan Dağıtık Hizmet Reddi (DDoS) saldırı trafiği ve CAIDA internet izlerinden oluşan CAIDA DDoS oluşturulmuştur. Bu veri seti, yüksek hızlı bir internet omurgası üzerinde koklanan CAIDA Equinix-Chicago monitöründen alınan pasif ağ trafiğinden oluşmaktadır. Bu veri kümeleri, dar bir saldırı türü kapsamına sahip oldukları için standart veri kümeleri olarak kullanılamazlar (Shiravi et al., 2012; Tavallaee et al., 2009).

LBNL

Lawrence Berkeley Ulusal Laboratuvarı Veri Kümesi (LBNL) veri seti, TCP, Kullanıcı Veribloğu İletişim Kuralları (UDP) ve ICMP protokol servisleri ile ağ trafiğinin izleri yakalanarak oluşturulmuştur. Yakalanan paketler, paket yükü ile ilgili bilgiden yoksundur ve veri kümesinin anonimleştirme sınırlaması vardır (Nechaev et al., 2010). Web, e-posta, ad hizmetleri ve ağ dosya hizmetleri gibi dahili ve harici uygulamalar için uygulanır.

KYOTO

Kyoto veri seti, Kyoto Üniversitesi tarafından bal küpleri yardımıyla 2009 yılında geliştirilmiştir. Bu, bal küpü kullanılarak ağ izlerinin yakalanması ile oluşturulduğundan, yalnızca bal küpüne yönelik saldırılar analiz edilmiştir. Veri seti, DNS ve posta trafiği gibi az miktarda gerçekçi kullanıcı davranışıyla analiz edilen ağ trafiğinden oluşur. Bunun bir sonucu olarak, veri seti, üretilen uyarıları azaltmak için gerekli olan ihmal edilebilir yanlış pozitiflerden oluşmaktadır (J. Song et al., 2011).

TWENTE

Twente veri seti, Sperotto (2019) tarafından Twente Üniversitesi'nde geliştirilmiştir (Sperotto et al., 2009). Veriler, bal küpü ağı tarafından auth/ident kullanılarak Net-Flow tarafından yakalandı. Açık Güvenli Kabuk (OpenSSH), Pro FTP ve Apache web sunucusu gibi hizmetleri kapsayan ağ verilerini içerir. Yakalanan veriler tamamen müdahaleci veya normal değildir. Ayrıca sürekli ICMP ve İnternet Aktarmalı Sohbet Protokolü (IRC) trafiğini de içerir. Oluşturulan veri kümesi, oluşturulan uyarılar arasında minimum bir korelasyon ilişkisi ile etiketlenir ve izinsiz girişlerin çeşitliliği ve hacminin dar bir kapsamına sahiptir.

UMASS

Massachusetts Üniversitesi, kablosuz uygulamalarından bazı ağ izlerini koklayarak UMASS veri setini geliştirmiştir. Veri seti, yalnızca bir TCP tabanlı bağlantının gözlemlendiği saldırı senaryosu dikkate alınarak oluşturulmuştur. Çok sınırlı sayıda saldırı ve ağ verisine sahip olduğu için izinsiz girişleri tespit etme veya önleme yeteneğine sahip değildir (Prusty et al., 2011).

ADFA

ADFA veri seti, New South Wales Üniversitesi tarafından eğitim ve doğrulama seti kullanılarak vektör başına on saldırı depolanarak geliştirilmiştir. Saldırılar, kaba kuvvet kullanarak FTP ve SSH şifre çalma, java yorumlayıcı, süper kullanıcı ekleyerek yönetici ayrıcalıklarını kötüye kullanma, Linux meterpreter ve C100 web shell saldırıları olarak sıralanabilir. Saldırılar, iyi huylu ağ trafiği tarafından iyi ayırt edilemez. Bu veri seti, farklı saldırı kategorilerini tanımlamaz (Xie & Hu, 2013).

4.2. Sınıflandırma Algoritmaları

Yapay Zeka (YZ) alanı, 1950'lerde bilgisayar bilimcilerinin bilgisayarların insanlar gibi "düşünüp düşünemeyeceğini" belirlemek için yola çıkmasıyla doğmuştur. Yapay Zeka, Massachusetts Teknoloji Enstitüsü (MIT)'den Marvin Minksy tarafından "insanlar tarafından yapıldığında zeka gerektirecek şeyleri makinelere yaptırma bilimi" olarak tanımlanmıştır (Boden, 1987). Yapay Zeka, Makine Öğrenmesi (ML) ve Derin Öğrenme (DL) tekniklerinin çatısı konumundadır. Chollet (2018) tarafından yapay zeka alanı için yapılan bir diğer benzer tanım ise basitçe "normalde insanlar tarafından gerçekleştirilen entelektüel görevleri otomatikleştirme çabası" şeklindedir (Chollet, 2018). Dolayısıyla, YZ yalnızca ML ve DL alt alanlarını değil, aynı zamanda normalde insanlar tarafından gerçekleştirilen entelektüel görevleri otomatikleştirme hedefini sağlamaya yönelik diğer birçok yaklaşımı da kapsamaktadır. Ancak bu diğer yaklaşımlar, bilgisayarın öğrenmesini sağlama görevini içermez. Sembolik yapay zeka olarak bilinen sistemler, insanlar tarafından açıkça programlanan kurallara dayanır. Bu sistemler satranç oynamak gibi iyi tanımlanmış mantıksal görevleri çözmede iyi performans gösterirler; ancak görüntü sınıflandırma ve dil çevirisi gibi daha karmaşık görevlerin üstesinden gelmek için yeterli donanıma sahip değildirler. Bu nedenle, makine öğrenmesi olarak adlandırılan yapay zeka yaklaşımı, önceki yaklaşımlara göre daha fazla ilgi görmeye başlamıştır.

Alan Turing'in 1950 yılında yayınladığı "Computing Machinery and Intelligence" isimli makalede, genel amaçlı bilgisayarların öğrenme yeteneğine sahip olduğu ve orijinallik özelliği taşıyabileceği sonucuna varılmıştır (Turing, 2009). Bu; bilgisayarların manuel olarak kuralları girmek yerine, verilere dayalı olarak kuralları öğrenip öğrenemeyeceği sorusunu ortaya çıkarmış ve makine öğrenimi alanının doğmasına yol açmıştır. Makine öğrenimi algoritmaları, veri üzerinden öğrenen ve

ayarlayan algoritmalardır. Bu algoritmalarda, bilgisayarın ne yapacağını açıkça söylemek yerine, örneklere ve verilere dayalı olarak programlamayı öğrenmesine izin verilir. Bu sayede bilgisayarlar, daha önce hiç görmedikleri yeni girdiler üzerinde kararlar verebilir ve görevleri yerine getirebilir.

Öğrenme algoritmasını tanımlayan Mitchell (1997), bir bilgisayar programının deneyim ile geliştiğini ve deneyim sayesinde belirli bir görevde performansının arttığını belirtmiştir (Mitchell, 1997). Bu çalışmada, görev bir ağ akışının iyi veya kötü huylu olarak sınıflandırılmasıdır. Her ağ akışı, belirli ölçümlerde bir dizi özelliğe sahiptir ve bu özellikler bir vektörde temsil edilir.

Çeşitli görev türleri olsa da makine öğrenimi algoritmalarının odaklandığı iki sınıflandırma görevi vardır; ikili ve çok sınıflı sınıflandırma. İkili sınıflandırmada, öğrenme algoritmasının amacı, bir girdiye karşılık gelen çıkış sınıfını belirlemektir. Çıkış sınıfı, ağ akışının iyi ya da kötü huylu olduğunu ifade eder. Çok sınıflı sınıflandırmada ise çıkış sınıfı, ağ akışının belirli bir saldırı kümesinden birine ait olduğunu ifade eder.

Makine öğrenimi algoritmaları, denetimli ve denetimsiz olmak üzere iki ana kategoriye ayrılabilir. Denetimli algoritmalar, önceden etiketlenmiş verilerle eğitilirken, denetimsiz algoritmalar, etiketlenmemiş veriler üzerinde yapılandırmalar yaparlar.

Bu bölümde bu araştırmada kullanılan çeşitli makine öğrenimi algoritmalarına odaklanılacaktır. Bilgisayarların büyük veri kümelerinden matematiksel modelleri uygulayarak otomatik olarak öğrenmesine olanak tanıyan tüm teknik ve algoritmalara hep birlikte makine öğrenmesi adı verilir (Ahmad et al., 2020). Bu başlık altında çalışmada kullanılan maline öğrenmesi algoritmaları anlatılmıştır.

4.2.1. Lojistik Regresyon

Lojistik regresyon, bir veya birden fazla bağımsız değişken ve kategorik bağımlı değişkenler içeren veri kümelerini analiz etmek için kullanılan makine öğrenimi sınıflandırma algoritmalarından biridir (Miguel-Hurtado et al., 2016). Doğrusal regresyon, çıktıyı sürekli sayısal olarak kullanırken lojistik regresyon, daha sonra iki veya daha fazla ayrık sınıfla eşleştirilebilen bir olasılık değeri döndürmek için aşağıda denklemi verilen lojistik sigmoid fonksiyonunu kullanarak çıktısını dönüştürür (Ng & Jordan, 2001). Ayrıca, lojistik regresyon modeli doğrusal fonksiyon yerine daha

karmaşık maliyet yani sigmoid fonksiyonunu kullanır (Park, 2013). Lojistik regresyon maliyet fonksiyonunu 0 ile 1 arasında sınırlar.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad \textbf{(4.1)}$$

Formülde, $\sigma(x)$ 0 ile 1 arasındaki çıktı (olasılık tahmini), x: fonksiyon girdisi ve e: doğal log tabanıdır.

4.2.2. Karar Ağacı

C4.5 karar ağacı algoritması sınıflandırma metodolojilerinde uygulama alanı bulur. Bu teknik çok sayıda kovaryansı ilişkilendirir, dolayısıyla bunlara benzer algoritmaların ve değişkenlerin tahminini geliştirir (Song et al., 2015). C4.5 sınıflandırıcısı, kararların ve bunların olası sonuçlarının ağaç benzeri bir modelini oluşturan bir tür denetimli makine öğrenimi algoritmasıdır. Model, giriş alanının yinelemeli olarak daha küçük bölgelere bölünmesiyle oluşturulur; burada her bölüm bir karar veya sınıf etiketiyle ilişkilendirilir. Karar ağacı oluşturma süreci, verileri her bir dahili düğümde bölmek için en bilgilendirici özelliğin seçilmesini içerir. Bir karar ağacı, örnek alanını yinelemeli olarak bölümlendirerek bir sınıflandırıcı görevi görür. Karar ağacının düğümleri, kök düğümün gelen kenarlarının olmadığı ve diğer düğümlerin her birinin yalnızca bir gelen kenarına sahip olduğu köklü bir ağaç oluşturur. İç düğümlere test düğümleri de denir ve bunların giden kenarları vardır, geri kalan düğümlere ise yaprak düğümler veya terminal düğümler denir. Karar ağacının dahili düğümleri, giriş öznitelik değerlerine bağlı olan ayrı bir işlevi kullanarak örnek alanını iki veya daha fazla alt alana böler. Genellikle her test tek bir özelliği dikkate alır ve örnek alanını özellik değerine göre böler. Nitelik sayısal ise, test koşulu bir değer aralığını belirtir (Rokach & Maimon, 2005).

4.2.3. Rastgele Orman

Rastgele orman, bir topluluk olarak işlev gören birden fazla bireysel karar ağacından oluşur. Sınıf tahmininin belirlenmesi, bireysel ağaçların sonucunu ilişkilendirecek, daha sonra topluluğu oluşturacak ve en yaygın olarak üzerinde mutabakata varılan görüşle sonuçlandırır (Wu et al., 2017). Rastgele orman sınıflandırıcı, yeni vakaları kategorize etmek için topluluk tekniği, çeşitli eğitimli sınıflandırıcılardan gelen göstergeleri birleştirir. Ağaçlar halinde yapılandırılmış sınıflandırıcılardan oluşan bir

sınıflandırıcıya rastgele orman adı verilir. Bağımsız rastgele vektörler bu sınıflandırıcılarda benzer şekilde dağıtılır. Ayrıca her ağaç en popüler sınıf lehine tek bir oy verir. Bir ağacın oluşturulması eğitim testi yoluyla sağlanırken, yeni bir rastgele vektör aynı dağılıma sahip önceki herhangi bir rastgele vektörle ilişkilendirilmez (Subasi, 2020).

RF, birçok özelliğe sahip (400'e kadar) büyük veri kümelerinde yüksek performans elde etme konusunda benzersiz bir özelliğe sahiptir. RF hem makine öğrenmesi hem de derin öğrenmede kullanılan çok yönlü bir sınıflandırıcıdır. Genellikle herhangi bir veri kümesindeki ilk denemelerde tercih edilen algoritmadır (Chen et al., 2022). RF'nin onu makine öğrenmesinde değerli bir algoritma haline getiren çeşitli güçlü yönleri vardır. İlk olarak, verilerin hızlı işlenmesine olanak tanıyan hızı ve verimliliği ile bilinmektedir. Ek olarak RF, dengesiz ve eksik verileri etkili bir şekilde işleme yeteneğine sahiptir ve bu da yüksek tahmin doğruluğu sağlar. Toplulukta birden fazla ağacın kullanılması, algoritmanın aykırı değerlerin etkisini önlemesine ve azaltmasına yardımcı olarak sağlamlığını artırır.

4.2.4. K-En Yakın Komşu

K-en yakın komşu, bir veri noktasının belirli bir gruba üye olma olasılığının diğer veri noktalarına yakınlığına bağlı olduğu parametrik olmayan denetimli bir öğrenme modelidir. Ait olduğu gruba göre sınıflandırma ve tahmin yapmak için veri noktasının yakınlığını değerlendirir. KNN, tutarlılığı ve çeşitli izinsiz giriş tespit problemlerine uygulanabilirliği nedeniyle popüler makine öğrenimi algoritmalarından biri olarak kabul edilir (Asharf et al., 2020).

KNN, sınıflandırma ve regresyon görevlerinde kullanılan basit ama etkili bir algoritmadır. Algoritmanın, eğitim verilerindeki k-en yakın komşularının sınıflarına veya değerlerine dayalı olarak yeni bir veri noktasının sınıfını veya değerini tahmin ettiği bir tür örneğe dayalı öğrenmedir. KNN, doğası gereği parametrik olmayan bir tür tembel öğrenme algoritmasıdır. Diğer algoritmalardan farklı olarak KNN, eğitim verileri sağlandığında herhangi bir eğitim gerçekleştirmez. Bu nedenle tembel öğrenen olarak da bilinir. Eğitim süresi boyunca KNN sadece verileri saklar ve herhangi bir hesaplama yapmaz. Yalnızca bir veri kümesi sorgusu çalıştırıldığında bir model oluşturur (Subasi, 2020).

KNN algoritması tarafından gerçekleştirilen sınıflandırmada, sonucu etkileyen üç faktör rol oynar: k değerinin ayarlanması, mesafe ölçümü için kullanılan yöntem ve uygulanan karar kuralları. K değerinin seçimi, tahmin sonucunu doğrudan etkilediği için çok önemlidir (G. Liu et al., 2022).

4.2.5. Saf Bayes

Saf Bayes, sınıflandırma görevi için yaygın olarak kullanılan bir olasılıksal makine öğrenme algoritmasıdır.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \textbf{(4.2)}$$

Yukarıda denklemi verilen Bayes teoremine göre formül, "B olayının meydana geldiği dikkate alındığında A olayının meydana gelme olasılığını" ifade etmektedir (Shafer, 1985). Bu bağlamda A bir hipotezi, B ise kanıtları temsil etmektedir. Saf Bayes algoritması, gözlemlenebilir sistem parametrelerine dayalı olarak her bilgisayar ağında meydana gelen bir saldırının olasılığını belirlemeyi amaçladığı için izinsiz giriş tespit problemleriyle ilgilidir.

Bu sınıflandırma tekniği, özellikle ağ trafiğinin normal veya saldırı olarak sınıflandırılan ikili senaryolarda önemlidir. Bayes teoremini uygulayan KDD CUP 99 veri seti üzerinde yapılan bir araştırmada, değerlendirme şu kategorilere ayrılan saldırılar için yaklaşık %85'lik bir doğruluk oranı ortaya çıkarmıştır (Sharmila & Nagapadma, 2019).

Saf Bayes için verilerin kategorik ve sayısal olmasına bağlı olarak farklı yaklaşımlar benimsenmesi gerekmektedir. STS'lerde kullanılan veri setlerinde hem kategorik hem de sayısal veriler bulunmaktadır. İki tip verinin de bulunduğu veri setleri üzerinde saf bayes ile sınıflandırma işleminin başarılı olması için veri ön işleme aşaması oldukça önemlidir. Veri önişleme aşamasında kategorik veriler one-hot encoding veya label encoding dönüşüm yöntemleri kullanılarak sayısal verilere dönüştürülür. Sonrasında sayısal veriler ile saf bayes yaklaşımı kullanılarak sınıflandırma işlemi gerçekleştirilir.

4.2.6. Gradyan Artırma ve Ekstra Gradyan Artırma

Artırma yaklaşımı, birden fazla zayıf model oluşturmayı ve performansı artırmak için bunları birleştirmeyi içerir. Hem gradyan artırma hem de ekstra gradyan artırma,

modellerin yeteneklerini geliştiren gradyan artırma (GB) ilkesini kullanır. Bu sınıflandırıcılar da alandaki diğer teknikler gibi denetimli öğrenme teknikleridir. Modern teknolojilerde genellikle Python yazılımı kullanılarak çok çekirdekli bilgisayarlarda uygulanırlar, ancak diğer programlama dillerinde de çalışabilirler. Bu algoritmaların kullanımı, özellikle eğitim ve test veri kümelerine uygulandığında sınıflandırma performansını artırır (Dadkhah et al., n.d.).

Genel olarak gradyan artırma, paralelleştirme, düzenlileştirme, doğrusal olmama, çapraz doğrulama ve ölçeklenebilirlik gibi önemli özellikler sunar.

4.2.7. Doğrusal Diskriminant Analizi

Doğrusal diskriminant analizi (LDA), daha düşük boyutlu bir alt uzayı tanımlamaya ve onu orijinal örnek uzayla karşılaştırmaya odaklanan bir veri analizi metodolojisidir. Amaç, veri kategorik özelliklerinde onları etkili bir şekilde ayıran doğrusal bir ilişki bulmaktır (Xanthopoulos & Pardalos, 2013).

LDA'nın uygulanması özellikle bilgi işlem sistemlerine yönelik ağ ve bağlantı noktası saldırıları bağlamında önemlidir. LDA, izinsiz giriş tespitine karşı mücadelede çok önemli bir makine öğrenimi algoritmasıdır (Lee, 2016). LDA kullanmanın bir avantajı, normal dağılım varsayımlarını çıkarım yapma, özelliklere ve tahminlere uygulama yeteneğidir. Ancak kategorik değişkenlerle etkili bir şekilde çalışmayabilir.

4.2.8. İkinci Dereceden Diskriminant Analizi

İkinci dereceden diskriminant analizi (QDA), LDA'nın aynı özelliklerini takip eder. Bununla birlikte sınıflandırmaya bağlı olarak farklı kovaryans matrislerine izin verir.

QDA sınıflandırıcı, Bayes diskriminant analizinden türetilmiştir ve sağlıklı ve sağlıklı olmayan kan numunelerini ayırt etmek için kan hücresi görüntülerini analiz etmeye yönelik sağlık hizmetleri de dahil olmak üzere çeşitli alanlarda uygulamalara sahiptir. Ayrıca ML' de bağlantı noktası taramasını ve DoS saldırılarını tanımlamak için kullanılmaktadır (Abdulhammed et al., 2019).

4.2.9. Uyarlanabilir Artırma

AdaBoost, zayıf sınıflandırıcıları uyarlayıp birleştirerek topluluklar oluşturmak için kullanılan öne çıkan bir artırma algoritmasıdır. Bu algoritmanın benzersiz özelliği, önemli sayıda zayıf öğreneni birleştirerek güçlü bir sınıflandırıcı oluşturabilmesidir.

AdaBoost'ta zayıf öğrenenler genellikle tekli kararlar veren karar ağaçlarıdır. Her zayıf öğrenciye, topluluk modeline katkısını etkileyen bir ağırlık atanır. Topluluğun genel tahmini, "yumuşak oylama" olarak bilinen bir süreçle belirlenir. Ancak, sonucun beklenen sonuçtan sapabileceği durumlar vardır ve bu gibi durumlarda, zayıf öğrenenlerin ağırlıkları eşit olarak dağıtılır ve bu da "zor oylama" olarak bilinen duruma yol açar (Peppes et al., 2021).

4.2.10. Sinir Ağları

Yapay sinir ağı olarak da isimlendirilen sinir ağı (NN), öğrenme sürecindeki bir insan beyninin çalışma mantığından etkilenilerek ortaya çıkan bir yaklaşımdır. İlk olarak 1943 yılında, insan beyninde bulunan hücrelerin yapısından yola çıkılarak matematiksel bir modellemesi oluşturularak geliştirilmiştir (McCulloch & Pitts, 1943). Makine öğrenmesinin bir dalı olan derin öğrenme, karmaşık sorunları çözmek için sinir ağlarını uygulamaya odaklanır. Sinir ağlarının uygulaması, bir zamanlar zaman alıcı ve karmaşık olan görevleri otomatikleştirmek için kullanıldıkları IoT cihazları, endüstriyel sektörler ve sağlık endüstrileri de dahil olmak üzere yaygındır (Hodo et al., 2016). Sinir ağlarının önemli faydalarından biri, ağları sürekli olarak izleme ve bilgi işlem sistemlerine olası izinsiz girişleri tahmin etme kapasitelerinde yatmaktadır; bu, günümüzün çok sayıda varlığa sahip yüksek derecede birbirine bağlı ortamında kritik bir husustur. Bu modeller, kötü amaçlı yazılım, DoS saldırıları, veri ihlalleri ve ağ hizmeti ihlalleri dahil olmak üzere çeşitli saldırı türlerini tanımlama yeteneğini gösterir. Ancak sinir ağı modellerinin eğitim sistemleri üzerindeki performansı, çoğu zaman ağ eğitimindeki sorunlardan dolayı bazen eksiklikler sergileyebilir. Tipik olarak %95 aralığında bildirilen doğruluk oranları nedeniyle yüksek doğruluk performansına ulaşmak zor olabilmektedir (P. Li et al., 2019). Veri kümesinin gereğinden fazla ayarlanması ve genelleme yeteneğinin zayıf olması, bu düşük performans ölçümlerinin ardındaki yaygın nedenlerdir. Bu zorlukların üstesinden gelmek için sinir ağlarını diğer modellerle birleştirmek çoğu zaman performansı artırabilmektedir.

Bir insanın öğrenme işlemi, yan yana duran sinapsların birbirleriyle bağlantıları aracılığıyla gerçekleşir (Hebb, 1949). Yapay sinir ağlarında yapay nöronlar bulunur. Yapay sinir ağının tamamı bir dizi katman halinde düzenlenmiş yapay nöronlardan oluşur. Sinir ağlarının karmaşıklığı, bir katmanın ister bir düzine birime ister

milyonlarca birime sahip olsun, veri kümesindeki temel modellerin karmaşıklığına bağlı olmaktadır. Yapay sinir ağlarında bulunan katmanlar genellikle giriş, çıkış ve gizli katmanladır. Giriş katmanında, öğrenilmesi veya analiz edilmesi gereken veriler dış dünyadan alınır.

Tam bağlantılı bir yapay sinir ağında, bir giriş katmanı ve birbiri ardına bağlanan bir veya daha fazla gizli katman bulunur. Her nöron, bir önceki katmandaki nöronlardan veya giriş katmanından girdi alır. Bir nöronun çıktısı, ağın bir sonraki katmanındaki diğer nöronların girdisi olur ve bu süreç, son katman ağın çıktısını üretinceye kadar devam eder. Daha sonra bu veriler bir veya daha fazla gizli katmandan geçtikten sonra çıkış katmanı için değerli verilere dönüştürülür. Son olarak çıktı katmanı, yapay sinir ağının gelen verilere verdiği yanıt şeklinde bir çıktı sağlar.

Sinir ağlarının büyük bölümünde birimler bir katmandan diğerine bağlanır. Bu bağlantıların her birinin, bir birimin diğerini ne kadar etkilediğini kontrol eden ağırlıkları vardır. Sinir ağı, bir birimden diğerine geçtikçe veriler hakkında giderek daha fazla şey öğrenir ve sonuçta çıktı katmanından bir çıktı üretir. Buradaki amaç, bir bilgisayar sisteminin de insan beynindeki öğrenmeden yola çıkarak, sinir hücrelerinin matematiksel modellemesi ile benzer bir yaklaşım sergilemesini sağlamaktır.

Derin öğrenme, veri karmaşık bir yapıya sahip olduğunda ve yüksek boyutluluk içerdiğinde yeni örneklere genelleme yapabilen bir makine öğrenimi alt alanıdır. Ayrıca doğrusal olmayan modellerin büyük veri kümeleri üzerinde ölçeklenebilir şekilde eğitilmesine olanak sağlar (Goodfellow et al., 2016). Bu özellikler, ağ saldırı tespiti alanında oldukça önemlidir çünkü hem büyük miktarda veriyle uğraşmak gerekmektedir hem de model, şu anda mevcut etiketli verilerde özel olarak temsil edilmeyen yeni saldırı biçimlerine genelleme yapabilmelidir.

Derin öğrenmenin kökeni 1940'lara kadar uzanmasına rağmen, son yıllarda popülerlik kazanmıştır. İlk algoritmalardan bazıları biyolojik olarak insan beyninin hesaplama modellerinden esinlenmiştir (Goodfellow et al., 2016). Ancak modern derin öğrenme, çoklu kompozisyon seviyelerine dayanan prensiplere dayanmaktadır. Derin öğrenme algoritmaları, genellikle en az 5.000 etiketli örnek içeren veri kümeleri ile iyi performans gösterir. İnsan performansını aşmak için ise en az 10 milyon etiketli örnekle eğitilmeleri gerekmektedir.

Ağ saldırı tespiti alanında kullanılan geçmişteki veri kümeleri genellikle daha küçük boyutludur. Ancak ISCX IDS 2012 ve CIC IDS 2017 gibi güncel veri kümeleri daha büyüktür. Derin öğrenme mimarilerinin, algoritmalarının etkinliğini denemek ve belirlemek için kullanılabilir. Bu veri kümeleri, sinir ağının öğrenebileceği daha fazla örnek içermektedir. Derin öğrenme teknolojileri, kurumsal ağlardaki ağ trafiği ve günlük verilerini analiz ederek kötü niyetli faaliyetleri tespit etmek için kullanılabilir. Kurumsal ağlar genellikle hem ağ akış verilerini hem de çeşitli günlük (log) verilerini içerir ve uzman analistler tarafından sağlanan geri bildirimlerle desteklenir.

4.3. Özellik Seçimi

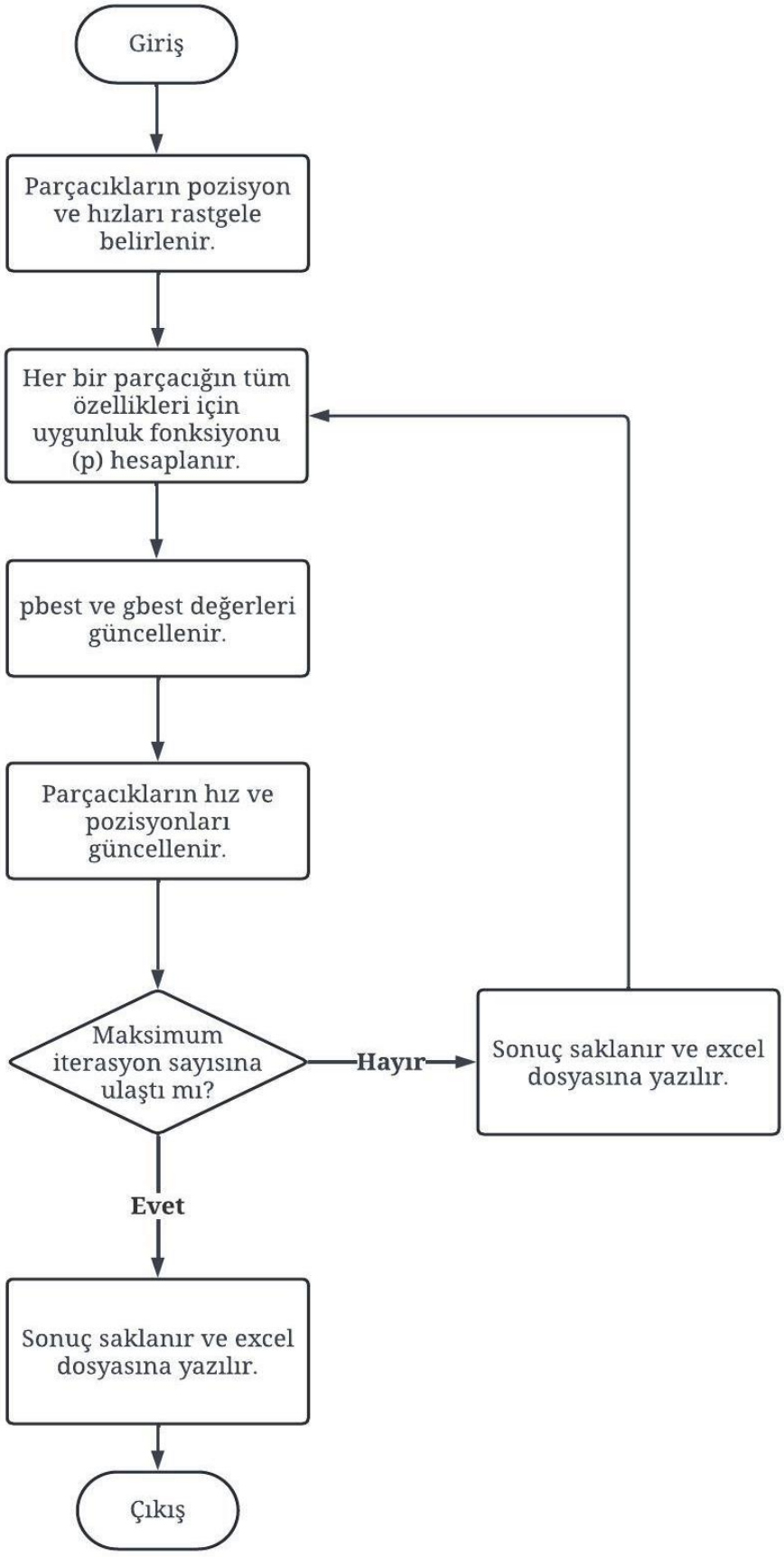
Bu çalışmada sınıflandırma problemlerinin performansını etkileyen özellik seçimi konusuna odaklanılmıştır. Performans değerlendirmesi için veri setlerinde bulunan özelliklerin seçiminde ve azaltılmasında meta-sezgisel yaklaşımlar kullanılmıştır. Bu yaklaşımlar aşağıda açıklanmıştır.

4.3.1. Parçacık Sürü Optimizasyonu

Parçacık Sürü Optimizasyonu algoritması, 1995 yılında R. Eberhart ve J. Kennedy tarafından önerilmiştir (Kennedy, 2010). Çeşitli mühendislik dallarında uygulanan PSO algoritması kuş ve balık sürülerinin davranışından türetilmektedir (Y. Zhang et al., 2015). Balıklar ve kuşlar gibi sosyal hayvanların hızla değişen etkileşimlerine ve hareketlerine uyum sağlamayı içerir. PSO algoritması, grup olarak birlikte çalışırken öğrenilen deneyimler ile kişisel deneyimleri birleştirir. Optimize edilmiş çözümler, kuşların sürü davranışıyla elde edilir. Kuşlar, belirlenmiş besin kaynaklarına ulaşmak için önceden belirlenmiş bir yolu izlerler. En kısa yol olarak kabul edilen bu yola aynı zamanda parçacığın kişisel en iyi çözümü (pbest) de denir. Her parçacık, kendi uçuş deneyimlerini ve gruptaki diğer kişilerin deneyimlerini gözlemleyerek arama uzayında en iyi çözümü arar. Bir başka en iyi uygunluk değeri, gruptaki herhangi bir parçacığın o parçacığın aralığına yakın gözlemlenmesiyle elde edilir. Buna global en iyi çözüm (gbest) denir. Her parçacığın pbest ve gbest' e ulaşmaya yönelik ivme için ilişkili hızı vardır. PSO' nun temel konsepti global optimal çözüme ulaşmak, böylece her parçacığı her adımda keyfi ağırlıkla pbest ve gbest' e doğru hareket ettirmektir. Bu algoritma gelişmiş keşif ve kullanım sağlamaktadır (Kumar & Ramakrishnan, 2016; Mirjalili et al., 2014).

PSO algoritması sürekli (continuous) uzayda parçacıkların hareket etmesi yaklaşımını benimsemektedir. STS için yapılan çalışmalar kesikli (discrete) problem olduğundan, PSO algoritmasının uyarlanması gerekir. Parçacıkların başlangıç pozisyonu ve hızları sürekli olarak belirlenir ve ardından eşik değeri belirlenerek ikili (binary) dönüşüm işlemi yapılır. İkili değerlere dönüştürülen parçacıkların uygunluk fonksiyonu hesaplanır. Ardından sürekli uzayda parçacıların güncel hız ve pozisyonları hesaplanır. Güncel hız ve pozisyonlar için ikili dönüşüm işlemi uygulanır ve en iyi çözüm güncellenir. Bu şekilde, PSO algoritması sürekli uzayda çalışmasına rağmen kesikli problemler için uyarlanmış olur.

PSO algoritmasının akış diyagramı şekil 4.1'de verilmiştir.



Şekil 4.1: PSO Akış Diyagramı.

Modelde kullanılan PSO algoritmasına ait akış diyagramı yukarıda verilmiştir. Burada başlangıç olarak 10 tane parçacık ve veri setinde bulunan özellik sayısı boyutunda popülasyon oluşturulur. Her bir parçacığın tüm özellikleri için başlangıç konumu ve hızları rastgele belirlenir. Bunların ikili (binary) dönüşümü yapılır. İkili dönüşüm bir eşik değeri (0,5) belirlenerek; eşik değerinden küçük olan 0, eşik değerinden büyük olan 1 olacak şekilde yazılır. Sonrasında aşağıda verilen 4.3 uygunluk fonksiyonu özellik seçimi veri seti üzerinden hesaplanır. Hata oranı, bu veri setinin k en yakın komşu algoritması kullanılarak sınıflandırma işlemi yapılmasının ardından elde edilen doğruluk oranının 1'den çıkarılması ile elde edilir. Bunun uygunluk fonksiyonuna ağırlığı %90 olarak belirlenmiştir. Seçilen özellik sayısının, toplam özellik sayısına oranının ağırlığı ise %10 olarak belirlenmiş ve uygunluk fonksiyonu hesaplanmıştır. Her bir parçacığın pbest değeri uygunluk fonksiyonu ile kıyaslanarak; eğer p değeri pbest değerinden daha iyi ise, p değeri pbest değerine atanır. Sonrasında tüm parçacıkların pbest değerleri kıyaslanarak, en iyi pbest değeri gbest değerine atanır. Her bir parçacık için 4.4'te verilen denkleme göre hız, 4.5'te verilen denkleme göre konum değerleri güncellenir. Bu aşamalar maksimum iterasyon sayısına ulaşana kadar tekrar edilir ve her bir iterasyonun gbest değeri excel dosyasına yazılır. 200 iterasyon arasından seçilen en iyi gbest değeri dikkate alınarak özellik seçimi yapılır. Seçilen özellikler ve özellik sayısı excel dosyasına yazılır.

$$p = \alpha * Hata\ Oranı + \beta * \frac{Seçilen\ Özellik\ Sayısı}{Toplam\ Özellik\ Sayısı} \quad \textbf{(4.3)}$$

p: Uygunluk Fonksiyonu α: 0,9, β: 0,1 olarak alınmıştır.

$$v_{i+1} = v_i + c_1 * rand_1 * (pbest - x_i) + c_2 * rand_2 * (gbest - x_i) \quad \textbf{(4.4)}$$

$$x_{i+1} = x_i + v_{i+1} \quad \textbf{(4.5)}$$

x: parçacık değeri, v : parçacığın değişim hızı, c_1,c_2 : sabit değerler, $rand_1$ ve $rand_2$: rastgele üretilen değerlerdir.

4.3.2. Çiçek Tozlaşma Algoritması

Bitkilerin çoğunluğu çiçekli bitkilerdir ve dünya çapında 250.000' den fazla çiçekli bitki türü vardır. Tozlaşma bitkilerin ana üreme stratejisini temsil etmektedir (Cronquist, 1981; Hutchings & Bell, 1991). Tozlaşma, polenlerin rüzgar ya da böcekler, kelebekler, arılar, kuşlar ve yarasalar gibi tozlayıcılar tarafından bir çiçekten diğerine aktarılması sürecidir. Çiçekli bitkiler tozlayıcıları çekmek ve tozlaşmayı sağlamak için nektar üretecek şekilde evrimleşmiştir (Glover, 2007). Çiçekli bitkilerdeki tozlaşmanın temeli aşağıda anlatıldığı gibidir:

- **Biyotik (global) tozlaşma:** Tozlaşmanın ana şekli, böcekler, kuşlar ve diğerleri gibi tozlaştırıcılar tarafından çapraz tozlaşma olarak da adlandırılan biyotik tozlaşmadır. Çiçekli bitkilerin neredeyse %90'ı bu tozlaşma şeklini kullanır. Polen taşıyıcıları çeşitli hızlarda hareket ettiklerinden ve hatta uçtuklarından, polenlerin hareketi oldukça uzun mesafeli olabilir. Bu tür tozlaşma, potansiyel levy uçuşu özelliklerine sahip küresel (global) tozlaşma olarak da düşünülebilir (Kalra & Arora, 2016; Pavlyukevich, 2007; X.-S. Yang, 2012). Polen bir çözüm vektörü olarak kodlanırsa, bu eylem küresel aramaya eşdeğer olabilir.
- **Abiyotik (lokal) tozlaşma:** Tozlaşmanın bir başka şekli de tozlaştırıcı gerektirmeyen, kendi kendine tozlaşma olarak da adlandırılan abiyotik tozlaşmadır. Oral bitkilerin yaklaşık %10'unun bu şekilde tozlaştığı tahmin edilmektedir. Tozlaşma yerel ve kendi kendine tozlaşma eğiliminde olduğundan, rüzgar ve difüzyon ile sağlanabilir (Glover, 2007; X.-S. Yang, 2012). Bu tür yerel hareketlerle kat edilen mesafe tipik olarak kısadır ve bu nedenle bu tür eylemler yerel arama olarak düşünülebilir.
- **Çiçek sabitliği:** Hem bitkiler hem de sinek kuşları gibi tozlayıcılar için, bazen başarı garantisi ile enerji tasarrufu sağlamak için bir ortaklık kurmak avantajlıdır. Sonuç olarak, çiçek sabitliği evrimleşmiştir. Bu durumda, tozlayıcılar yeni çiçek türlerini keşfetmek için enerji harcamadan yalnızca belirli bir çiçek türü kümesini ziyaret ederken, çiçek bitkileri tozlayıcıların sık ziyaretlerini teşvik etmek ve böylece üreme başarılarını en üst düzeye çıkarmak için tozlayıcılara yeterli nektar ödülü sağlayacak şekilde evrimleşir (Glover, 2007; Hutchings & Bell, 1991).

Yukarıdaki özellikler, çiçek tozlaşma algoritması (FPA) adı verilen bir optimizasyon algoritması tasarlamak için kullanılmıştır. 2012 yılında Xin-She Yang tarafından; tozlaşmanın temel özelliklerine dayanarak, çiçek tozlaşma algoritması geliştirilmiştir (X.-S. Yang, 2012).

Dört idealleştirme kuralı aşağıdaki gibi özetlenebilir:

- **Kural 1:** Global tozlaşma, tozlayıcıların polenleri Levy uçuşlarına göre taşıdığı biyotik ve çapraz tozlaşmayı içerir.
- **Kural 2:** Lokal tozlaşma, abiyotik ve kendi kendine tozlaşmayı içerir.
- **Kural 3:** Çiçek sabitliği, herhangi iki çiçek arasındaki benzerlikle orantılı bir üreme olasılığı olarak düşünülebilir.
- **Kural 4:** Geçiş olasılığı p ∈ [0,1], rüzgar gibi bazı dış faktörler nedeniyle yerel tozlaşma ve küresel tozlaşma arasında kontrol edilebilir. Lokal tozlaşma, toplam tozlaşma faaliyetlerinin önemli bir bölümüne sahiptir.

Yukarıda belirtildiği gibi, kuşlar ve yarasalar gibi tozlayıcılar biyotik tozlaşma sırasında polenleri uzun mesafelere taşıyabilir ve üreme için çeşitliliği ve en iyi tozlaşmayı sağlayabilir. Bu nedenle, birinci ve üçüncü FPA kuralları matematiksel olarak aşağıdaki gibi formüle edilebilir:

$$x_i^{t+1} = x_i^t + L(x_i^t - gbest) \quad \textbf{(4.6)}$$

Burada x_i^t iterasyonundaki polen veya çözüm vektörüdür ve gbest mevcut iterasyondaki tüm çözümler arasında bulunan en iyi çözümdür. L parametresi, aslında bir adım boyutu olan tozlaşmanın gücüdür. Tozlayıcılar çeşitli mesafe aralıklarıyla uzun mesafeler boyunca hareket ettiğinden, Levy uçuşu bu özellik için iyi bir simülatör olabilmektedir (X.-S. Yang, 2012). L parametresi aşağıdaki gibi bir Levy dağılımından hesaplanabilir:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, s \gg s_0 > 0 \quad \textbf{(4.7)}$$

Burada $\Gamma(\lambda)$ standart gama fonksiyonunu gösterir ve bu dağılım s>0 şartını sağlayan adımlar için geçerlidir. Normalde λ=1,5 kullanılması tavsiye edilmektedir. (X.-S. Yang, 2012).

Lokal tozlaşma herhangi bir tozlaştırıcı olmadan rüzgar veya difüzyon yoluyla gerçekleştiğinden, yerel tozlaşma ve çiçek sabitliği aşağıdaki gibi gösterilebilir:

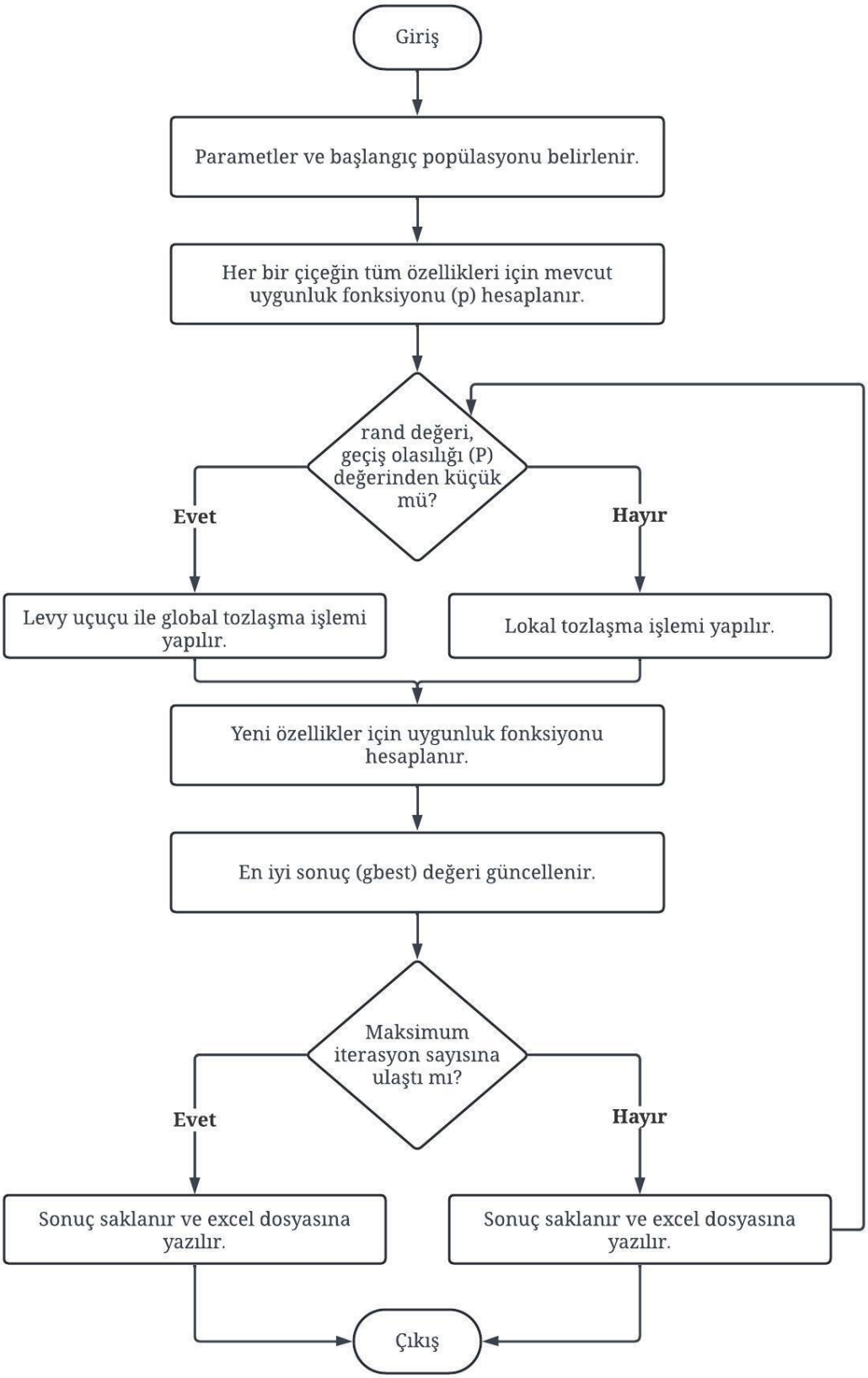
$$x_i^{t+1} = x_i^t + \mathcal{E}(x_j^t - x_k^t) \quad \textbf{(4.8)}$$

Burada x_j^t ve x_k^t aynı bitki türüne ait farklı çiçeklerden gelen polenlerdir. Bu denklem aslında sınırlı bir komşuluktaki çiçek sabitliğini taklit eder. Matematiksel olarak ifade etmek gerekirse, eğer x_j^t ve x_k^t aynı popülasyondan seçilebilecek aynı türlere aitse, [0, 1]'deki düzgün bir dağılımdan çekersek denklem yerel bir rastgele yürüyüş haline gelecektir ve üretilen yeni çözüm vektörü mevcut çözümlerden çok uzak olmayacaktır.

Yukarıda verilen bilgilere istinaden hem biyotik hem de abiyotik tozlaşma simüle edilmiştir fakat, her bir tozlaşma türünün yüzdesi ve sıklığı dikkate alınmamıştır. Bu özelliği taklit etmek için, P değerinin çözüm modifikasyonunun lokal veya global tozlaşmayı takip edip etmediğini belirlediği bir geçiş olasılığı kuralı kullanılır. Naif bir P=0.5 değeri kullanılabilse de daha gerçekçi ve etkili bir P=0.8 değeri çoğu uygulama için daha iyi performans sağlar (X.-S. Yang, 2012).

FPA algoritması sürekli uzayda bitkilerin tozlaşma sürecini modelleyen bir sürekli optimizasyon algoritmasıdır. STS için yapılan çalışmalar kesikli (discrete) problem olduğundan, FPA algoritmasının uyarlanması gerekir. Sürekli uzayda oluşturulan başlangıç popülasyonu için eşik değeri belirlenerek ikili dönüşüm işlemi gerçekleştirilir. İkili değerlere dönüştürülen her bir çiçeğin tüm özellikleri için uygunluk fonksiyonu hesaplanır. Ardından sürekli uzayda lokal ve global tozlaşma gerçekleştirilir. Elde edilen sonuçlar için ikili dönüşüm işlemi uygulanır ve en iyi çözüm güncellenir. Bu şekilde, FPA algoritması sürekli uzayda çalışmasına rağmen kesikli problemler için uyarlanmış olur.

Aşağıda bulunan şekil 4.2'de modelin akış diyagramı verilmiştir.



Şekil 4.2: FPA Algoritması Akış Diyagramı.

Burada başlangıç olarak 10 tane çiçek ve veri setinde bulunan özellik sayısı boyutunda popülasyon oluşturulur. Her bir çiçeğin tüm özellikleri için başlangıç konumu ve hızları rastgele belirlenir. Bunların ikili (binary) dönüşümü yapılır. İkili dönüşüm bir eşik değeri (0,5) belirlenerek; eşik değerinden küçük olan 0, eşik değerinden büyük olan 1 olacak şekilde yazılır. Sonrasında (4.3)'de verilen uygunluk fonksiyonu özellik seçimi veri seti üzerinden hesaplanır. Hata oranı, bu veri setinin k en yakın komşu algoritması kullanılarak sınıflandırma işlemi yapılmasının ardından elde edilen doğruluk oranının 1'den çıkarılması ile elde edilir. Bunun uygunluk fonksiyonuna ağırlığı %90 olarak belirlenmiştir. Seçilen özellik sayısının, toplam özellik sayısına oranının ağırlığı ise %10 olarak belirlenmiş ve uygunluk fonksiyonu hesaplanmıştır. İlk iterasyon için uygunluk fonksiyonu değeri kıyaslanarak en iyi sonuç değeri (gbest) güncellenir. Ardından her bir özellik için geçiş olasılığı değeri (P), rastgele elde edilen rand değeri ile kıyaslanır. "rand<P" ise levy uçuşu ile global tozlaşma işlemi yapılır. "rand>P" ise lokal tozlaşma işlemi yapılır. Elde edilen yeni özellikler için uygunluk fonksiyonu hesaplanır ve gbest değeri güncellenir. Bu aşamalar maksimum iterasyon sayısına ulaşana kadar tekrarlanır ve her iterasyonun gbest değeri excel dosyasına yazılır. 200 iterasyon arasından seçilen en iyi gbest değeri dikkate alınarak özellik seçimi yapılır. Seçilen özellikler ve özellik sayısı excel dosyasına yazılır.

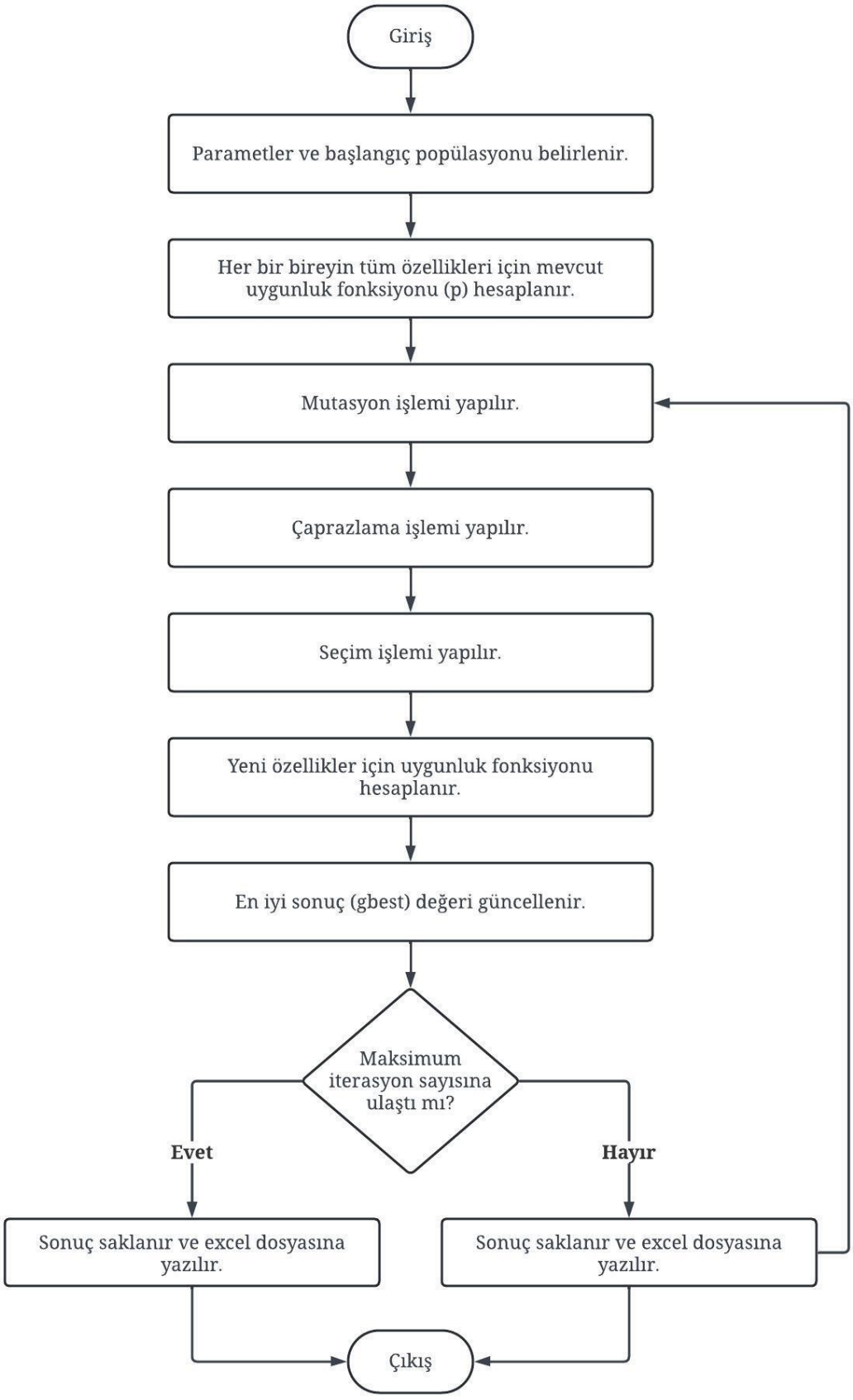
4.3.3. Diferansiyel Evrim

Diferansiyel evrim (DE), Storn ve Price (1997) tarafından bir optimizasyon arama tekniği olarak önerilen önemli bir evrimsel algoritmadır (Storn & Price, 1997). Evrimsel algoritmalara (EA) benzer şekilde DE, çaprazlama, mutasyon ve seçimi üç strateji olarak kullanır ve her nesilde bu stratejileri kullanarak global optimum çözüme ulaşır. Ayrıca, günümüzde kullanılan en verimli evrimsel algoritmalardan biridir. DE'de popülasyondaki her birey bir hedef vektör olarak adlandırılır. Popülasyondaki diğer kişilerin farklı vektörlerinden yararlanılarak, hedef vektörü bozan bir mutant vektör oluşturmak için mutasyon kullanılır. Çaprazlama prosedürü daha sonra mutasyon vektörünün parametrelerini popülasyondan seçilen bir ebeveyn vektörün parametreleriyle karıştırarak deneme vektörünü oluşturur. Son olarak, seçim işlemi, oluşturulan iz vektörleri ile ilişkili ebeveyn vektörleri arasında bire bir yazışma kurarak uygunluk değerine göre bir sonraki nesil için hangi vektörlerin seçilmesi gerektiğine karar verir.

Sezgisel bir optimizasyon tekniği olan DE yönteminin temel konsepti, gruplar arası bilgi alışverişi ve her bireyin kendine özgü hafızası aracılığıyla dinamik olarak global optimum değeri aramaktır. Popülasyon sürekli olarak; mutasyon, çaprazlama ve seçim süreçleri yoluyla optimum değere doğru gelişmektedir.

DE algoritması, sürekli uzayda çalışan bir evrimsel algoritmadır. STS için yapılan çalışmalar kesikli (discrete) problem olduğundan, DE algoritmasının uyarlanması gerekir. Sürekli uzayda oluşturulan başlangıç popülasyonu için eşik değeri belirlenerek ikili dönüşüm işlemi gerçekleştirilir. İkili değerlere dönüştürülen her bir bireyin tüm genleri için uygunluk fonksiyonu hesaplanır. Ardından sürekli uzayda mutasyon ve çaprazlama gerçekleştirilir. Elde edilen sonuçlar için ikili dönüşüm işlemi uygulanır ve en iyi çözüm güncellenir. Bu şekilde DE algoritması sürekli uzayda çalışmasına rağmen kesikli problemler için uyarlanmış olur.

Kullanılan DE algoritmasının akış diyagramı şekil 4.3'teki gibidir.



Şekil 4.3: DE Algoritması Akış Diyagramı.

Burada başlangıç olarak 10 birey ve veri setinde bulunan özellik sayısı boyutunda kromozom popülasyonu oluşturulur.

$$X_G = \{x_{1,G}, x_{2,G}, x_{3,G}, \cdots, x_{i,G}\} \quad \textbf{(4.9)}$$

Burada her kromozom $x_{i,G}$ hedef vektör olarak adlandırılır.

Her bir bireyin tüm özellikleri için başlangıç konumu rastgele belirlenir. Bunların ikili (binary) dönüşümü yapılır. İkili dönüşüm bir eşik değeri (0,5) belirlenerek; eşik değerinden küçük olan 0, eşik değerinden büyük olan 1 olacak şekilde yazılır.

Sonrasında (4.3)'de verilen uygunluk fonksiyonu özellik seçimi veri seti üzerinden hesaplanır. Hata oranı, bu veri setinin k en yakın komşu algoritması kullanılarak sınıflandırma işlemi yapılmasının ardından elde edilen doğruluk oranının 1'den çıkarılması ile elde edilir. Bunun uygunluk fonksiyonuna ağırlığı %90 olarak belirlenmiştir. Seçilen özellik sayısının, toplam özellik sayısına oranının ağırlığı ise %10 olarak belirlenmiş ve uygunluk fonksiyonu hesaplanmıştır. İlk iterasyon için uygunluk fonksiyonu değeri kıyaslanarak en iyi sonuç değeri (gbest) güncellenir.

Mutasyon işlemine geçilir ve burada, her $x_{i,G}$ için bir donör vektör $v_{i,G+1}$ oluşturulur.

$$v_{i,G+1} = x_{r1,G} + F.(x_{r2,G} - x_{r3,G}) \quad \textbf{(4.10)}$$

Burada $x_{r1,G}$, $x_{r2,G}$, $x_{r3,G}$ hedef vektör hariç popülasyondan rastgele seçilmiştir. F, diferansiyel varyasyonun ($x_{r2,G} - x_{r3,G}$) amplifikasyonunu kontrol eden kullanıcı tanımlı bir mutasyon veya sabit faktördür (F = 0.5).

Ardından çaprazlama işlemine geçilir. Donör yani mutant vektör, hedef vektör ile karıştırılarak aşağıdaki gibi bir deneme vektörü $u_{i,G+1}$ üretilir:

$$u_{i,G+1} = \begin{cases} v_{j,i,G+1} \; if \; rand_{j,i} \leq CR \; or \; j = I_{rand} \\ x_{j,i,G} \; if \; rand_{j,i} > CR \; and \; j \neq I_{rand} \end{cases} \quad \textbf{(4.11)}$$

$i = 1,2, \dots, N$; $j = 1,2, \dots, D$, burada N popülasyon boyutu ve D x_i'nin boyutudur. CR çaprazlama sabitidir (CR=0.9). I_{rand}; $v_{i,G+1} \neq x_{i,G}$ olmasını sağlar.

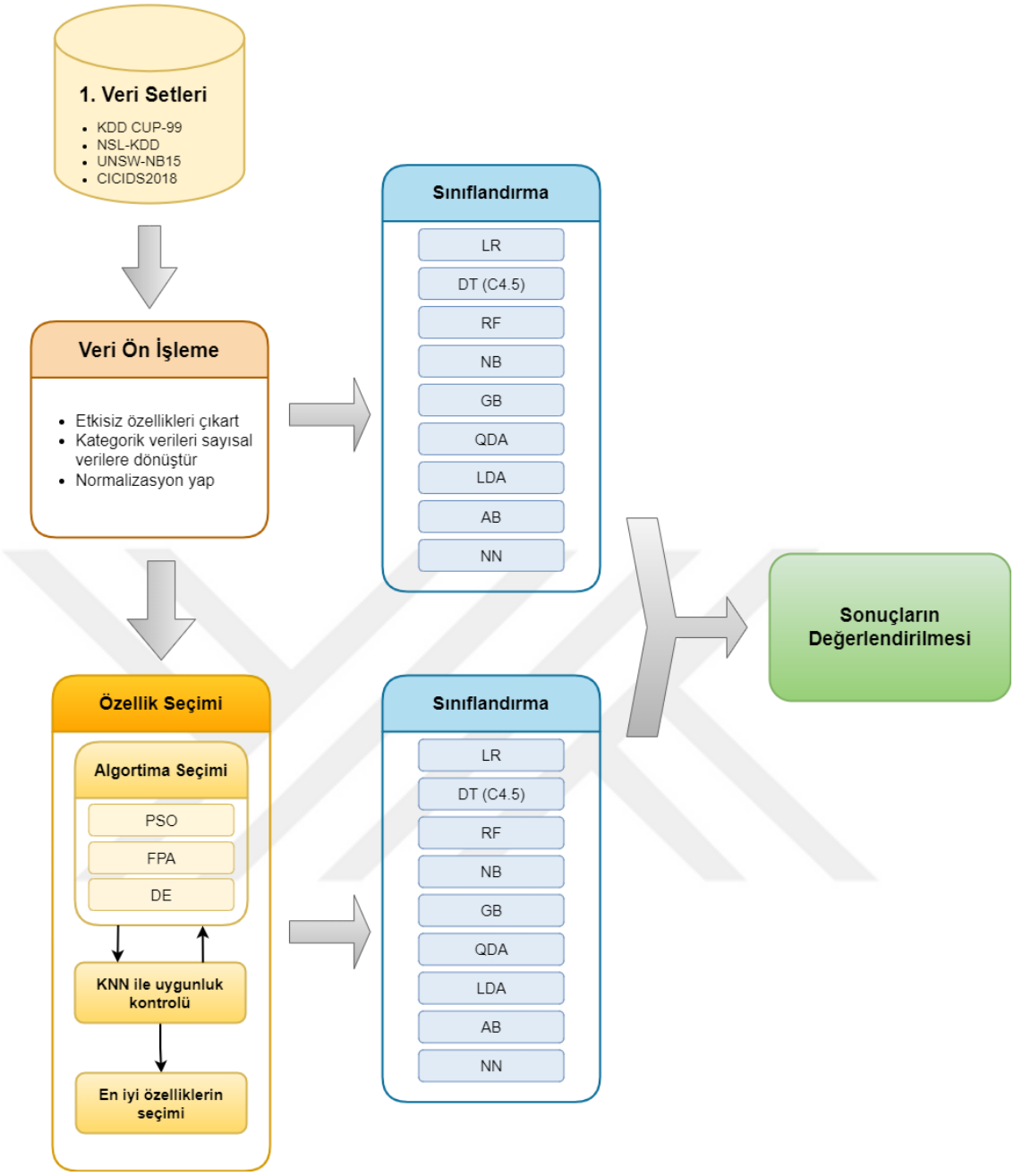
Çaprazlama işlemi tamamlandıktan sonra seçim işlemi yapılır. Deneme vektörü $u_{i,G+1}$ ve hedef vektör $x_{i,G}$ uygunluk fonksiyonu kullanılarak değerlendirilir. Daha sonra ikisi arasında bir karşılaştırma yapılır. Değeri düşük olan yeni nesle aktarılır.

$$x_{i,G+1} = \begin{cases} u_{i,G+1} \; if \; f(u_{i,G+1}) \leq f(x_{i,G}) \\ x_{i,G} \; diğer \; durumlarda \end{cases} \quad \textbf{(4.12)}$$

Elde edilen yeni özellikler için uygunluk fonksiyonu hesaplanır ve gbest değeri güncellenir. Bu aşamalar maksimum iterasyon sayısına ulaşana kadar tekrarlanır ve her iterasyonun gbest değeri excel dosyasına yazılır. 200 iterasyon arasından seçilen en iyi gbest değeri dikkate alınarak özellik seçimi yapılır. Seçilen özellikler ve özellik sayısı excel dosyasına yazılır.

4.4. Önerilen Model

Bu çalışmada saldırı tespit sistemlerinde kullanılan makine öğrenmesi algoritmalarının performansını artırmak için meta sezgisel algoritmalar ile özellik seçimi önerilmiştir. Veri setleri üzerinde öncelikli olarak sınıflandırma algoritmaları ile testler yapılmıştır. Sonrasında ayrı olarak özellik seçimi algoritmaları ile sınıflandırma testleri yapılmıştır. Tüm yapılan terslerin sonuçları karşılaştırmalı olarak analiz edilmiştir. Önerilen modelin akış diyagramı aşağıda şekil 4.4'te gösterilmektedir.



Şekil 4.4: Önerilen Modelin Akış Diyagramı.

Modelde kullanılan veri setleri, öncelikli olarak rastgele %20-80 oranında özellik seçimi eğitim ve sınıflandırma veri seti olarak ayrılmıştır. Bu veri setleri sırasıyla kod tarafından okunur. Sonrasında veriler ön işleme ve normalizasyon işlemine sokulur. Ön işleme aşamasında; sıra numarası, atak kategorisi vb. sonuca etkisi olmayacak özellikler çıkartılır ve protokol, servis vb. gibi kategorik veriler sayısal verilere dönüştürülür. Dönüştürme işlemi label encoding tekniği kullanılarak gerçekleştirilir. Normalizasyon işlemi, veri setinde bulunan her özelliğin kendi değerinden, değerlerin ortalamasının çıkarılması ve standart sapmaya bölünerek elde edilir. Standart

ölçekleme (standard scaler) yöntemi kullanılarak normalizasyon işlemi geçekleştirilir. Standart ölçekleme yöntemi, veri setinde bulunan her özelliğin ortalamasını 0 ve standart sapmasını 1 olacak şekilde dönüştürür. Bu işlem sayesinde farklı ölçeklerin özellikler üzerindeki etkisi azaltılarak, makine öğrenmesi modellerinin daha iyi performans elde etmesi sağlanır. Bu işlemden sonra, veriler üzerinde meta sezgisel; parçacık sürü optimizasyonu, diferansiyel evrim ve çiçek tozlaşma algoritmalarından biri kullanılarak özellik seçimi gerçekleştirilir. Seçilen özelikler ile sınıflandırma işlemine geçilir. Sınıflandırma işlemi için ayrılan veri seti, rastgele %80 eğitim ve %20 test verisi olacak şekilde ayrılır. Seçilen özellikler ile modelde; lojistik regresyon, karar ağacı, rastgele orman, lineer diskriminant analiz, kuadratik diskriminant analiz, adaboost, k en yakın komşu, saf bayes, gradyan artırma ve sinir ağları makine öğrenmesi algoritmaları sırasıyla uygulanarak sınıflandırma işlemi gerçekleştirilir. Her algoritmanın performans değerleri çapraz doğrulama yöntemi ile elde edilir. Elde edilen tüm sonuçlar excel dosyasına yazdırılır. Model bu işlemi 10 kez tekrarlar ve sonuçlar kaydedilir.

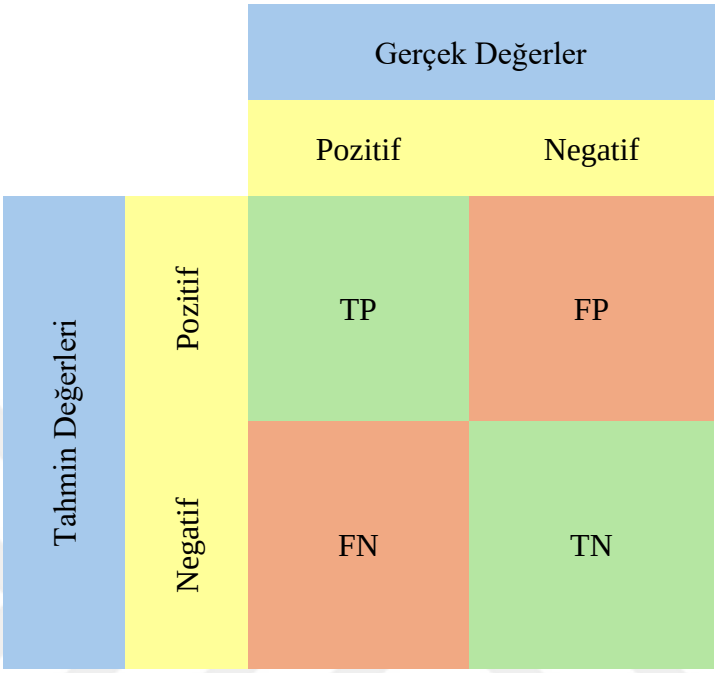
4.5. Değerlendirme Ölçütleri

Makine öğrenimi modelleri farklı değerlendirme ölçütleri (performans metrikleri) kullanılarak değerlendirilir (Kelleher et al., 2020). Tablo 4.9'da çeşitli ML modellerinin analizinde kullanılan performans metriklerini göstermektedir. Bu ölçümlerin hesaplanmasında karışıklık matrisi kullanılır. Şekil 4.5'te, karışıklık matrisinin ayrıntılı bir değerlendirmesini göstermektedir. Aşağıda kullanılan metriklerin genel tanımı verilmiştir.

- **Doğru Pozitif (True Positives, TP):** Bunlar, gerçek sınıf değerinin 'evet' olduğunu ve tahmin edilen sınıf değerinin de 'evet' olduğunu gösteren doğru pozitif tahminleri ifade eder.
- **Doğru Negatif (True Negatives, TN):** Bunlar, gerçek sınıf değerinin 'hayır' olduğunu ve tahmin edilen sınıf değerinin de 'hayır' olduğunu gösteren, doğru tahmin edilen negatif değerlerdir.

Gerçek sınıfın tahmin edilen sınıfla uyuşmadığı durumlarda yanlış pozitif ve yanlış negatif değerleri ortaya çıkmaktadır.

- **Yanlış Pozitif (False Positives, FP):** Yanlış pozitif, modelin pozitif kategoriye giren yanlış bir tahmin yaptığı durumu ifade eder.
- **Yanlış Negatif (False Negatives, FN):** Yanlış negatif, modelin negatif kategoriye ait hatalı bir tahminde bulunmasıdır.



Şekil 4.5: Karışıklık Matrisi.

Kelleher, (2020)'den uyarlanmıştır.

Tablo 4.9 da formülleri verilen performans ölçütleri aşağıda açıklanmıştır.

- **Doğruluk (Accuracy - acc):** Doğruluk, doğru şekilde sınıflandırılmış veri örneklerinin sayısını toplam örnek sayısına bölerek hesaplanan, doğru şekilde sınıflandırılmış veri örneklerinin oranını gösteren bir ölçüdür.
- **Kesinlik (Precision - pre):** Kesinlik, tahmin edilen tüm pozitif sonuçlar arasında doğru tahmin edilen pozitif sonuçların oranını ifade eder.
- **Hassasiyet (Sensitivity/Recall - rec):** Hassasiyet, tüm gözlemler dikkate alınarak, gerçek sınıftaki doğru tahmin edilen olumlu sonuçların oranının bir ölçüsüdür.
- **F1 Skoru (F1 Score – f1):** F1 Skoru, ağırlıklı ortalamalarını alarak hem Hassasiyet'i hem de Kesinlik'i dikkate alan bir ölçüdür. Hem yanlış pozitifleri hem de yanlış negatifleri hesaba katar.

Tablo 4.9 Performans Değerlendirme Ölçütleri.

Değerlendirme Ölçütleri	
Doğruluk	$\frac{TP + TN}{TP + TN + FP + FN}$
Kesinlik	$\frac{TP}{TP + FP}$
Hassasiyet	$\frac{TP}{TP + FN}$
F1 Skor	$\frac{2TP}{2TP + FP + FN}$

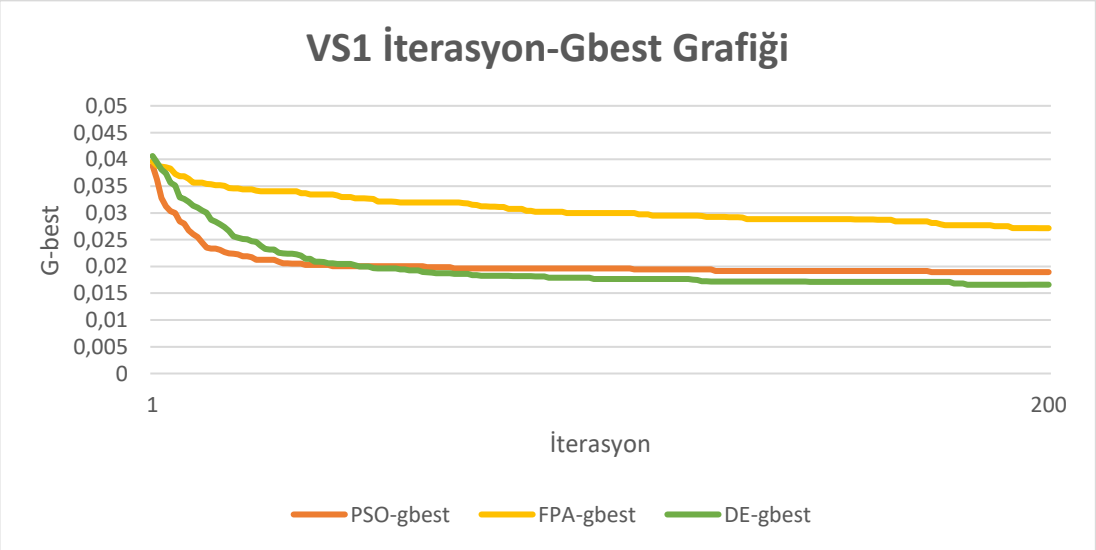
Kelleher, (2020)'den uyarlanmıştır.

5. DENEYSEL ÇALIŞMALAR

Bu bölümde deneysel çalışmaların sonuçlarına yer verilmiştir. Önerilen model 4.4 bölümünde anlatıldığı gibi; veri setleri öncelikli olarak özellik seçimi işlemi yapılmadan makine öğrenmesi algoritmaları kullanılarak sınıflandırma işlemine tabi tutulup, değerlendirme ölçütleri kullanılarak analiz edilmiştir. Bu yapılan sınıflandırma işlemi, SS olarak ifade edilmektedir. Sonrasından veri setlerine sırasıyla özellik seçimi için kullanılan PSO, FPA ve DE algoritmaları ayrı ayrı uygulanır. Bu adım sonrasında seçilen özellikler ile ayrı ayrı ve sırasıyla makine öğrenmesi algoritmaları sınıflandırma amacıyla uygulanıp değerlendirme ölçütleri kullanılarak analiz edilmiştir. Her bir veri seti için yapılan çalışmaların sonuçları aşağıda başlıklar altında anlatılmıştır.

5.1 KDD CUP 99 ile İlgili Çalışmalar

VS1 olarak ifade edilen KDD Cup 99 veri seti için yapılan çalışmalar; süre, doğruluk, kesinlik, hassasiyet ve F1 skor ölçütleri kullanılarak analiz edilmiştir. Bunlar aşağıda sırasıyla tablo ve şekillerde verilmiştir. Ayrıca PSO, FPA ve DE algoritmalarının 200 iterasyon için gbest değerleri aşağıda bulunan şekil 5.1'de verilmiştir.



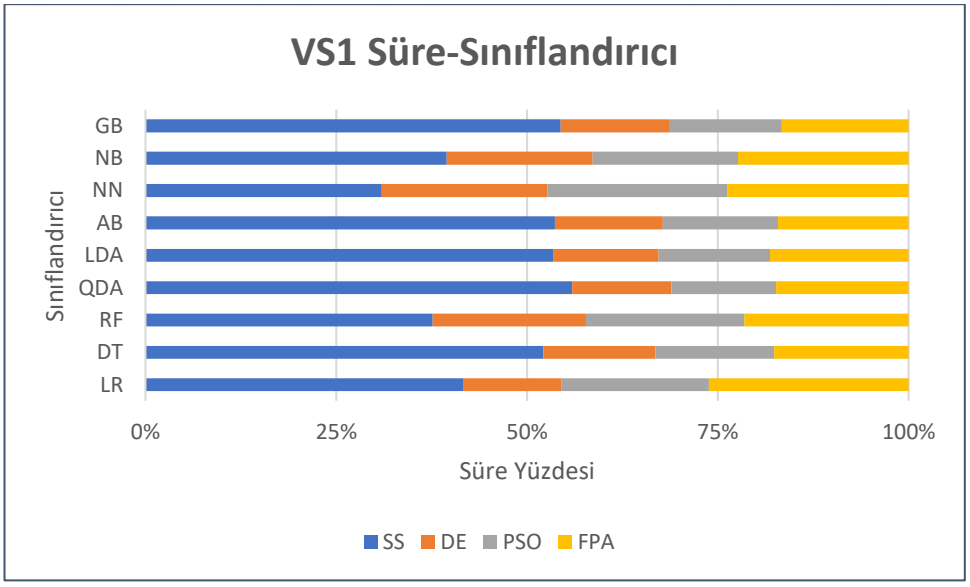
Şekil 5.1: VS1 İterasyon-Gbest Sonuçları.

Şekil 5.1'de görüldüğü üzere, yaklaşık 180. iterasyondan sonra FPA, PSO ve DE için en iyi gbest değeri elde edilmiştir.

Süre için analiz sonuçları; tablo 5.1 ve şekil 5.2'de verilmiştir.

Tablo 5.1: VS1 Süre Sonuçları.

VS1-Süre				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	10,5114	3,264421	4,866419	6,616302
DT	2,992934	0,842989	0,893131	1,011241
RF	36,3243	19,4011	20,03304	20,78567
QDA	3,969459	0,921591	0,975518	1,231699
LDA	5,025856	1,29495	1,370018	1,707212
AB	66,29612	17,4471	18,67448	21,14707
NN	136,1418	95,74716	103,5472	104,619
NB	1,646334	0,798128	0,79499	0,931854
GB	102,2422	26,82663	27,5909	31,23356



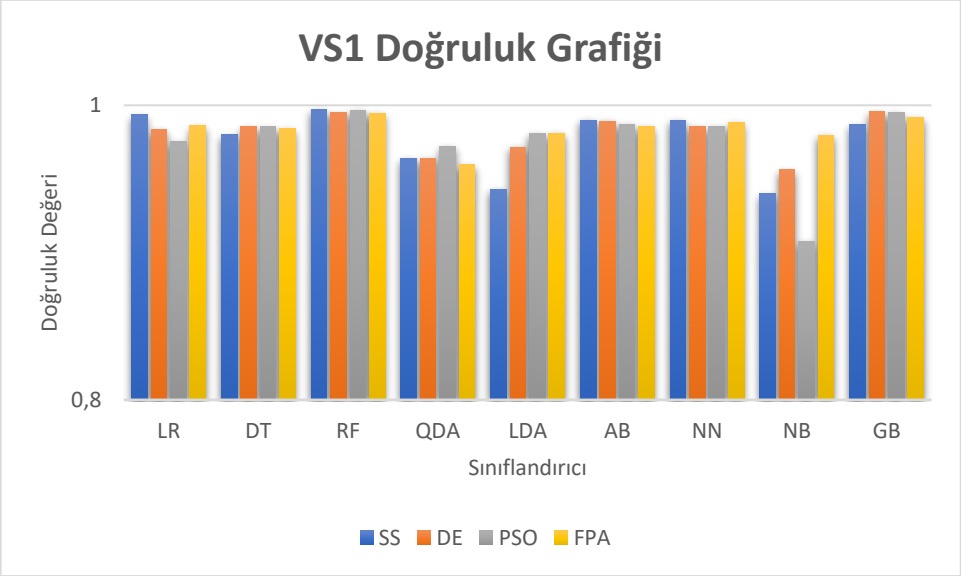
Şekil 5.2: VS1 Süre-Sınıflandırıcı Sonuçları.

Yukarıda verilen tablo ve grafik incelendiğinde, SS sonuçlarına göre özellik seçimi yaklaşımları kullanıldığında sınıflandırma süreleri 3'te 1 oranına gerileyerek süre açısından performansının arttığı görülmüştür.

Doğruluk ölçütü için analiz sonuçları; tablo 5.2 ve şekil 5.3'te verilmiştir.

Tablo 5.2: VS1 Doğruluk Sonuçları.

VS1-Doğruluk				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,993543	0,983386	0,975181	0,986195
DT	0,979991	0,985497	0,985514	0,984062
RF	0,996941	0,994775	0,996597	0,994623
QDA	0,963567	0,964215	0,971896	0,959556
LDA	0,94316	0,971441	0,980596	0,980887
AB	0,989897	0,989239	0,987167	0,985537
NN	0,989851	0,9856	0,985638	0,988004
NB	0,940063	0,95633	0,907571	0,979798
GB	0,986757	0,995529	0,994985	0,991675



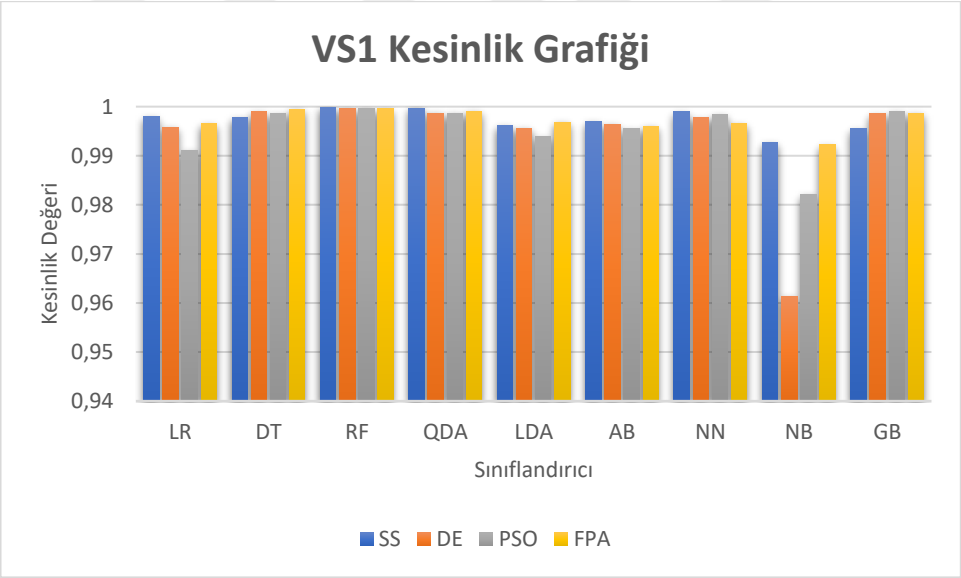
Şekil 5.3: VS1 Doğruluk-Sınıflandırıcı Sonuçları.

Doğruluk ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE algoritmasının DT, QDA, LDA, NB, GB' de, PSO algoritmasının DT, RF, QDA, LDA, GB'de, FPA algoritmasının DT, LDA, NB, GB'de daha yüksek doğruluk oranına sahip olduğu görülmüştür.

Kesinlik ölçütü için analiz sonuçları; tablo 5.3 ve şekil 5.4'te verilmiştir.

Tablo 5.3: VS1 Kesinlik Sonuçları.

VS1-Kesinlik				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,997991	0,995821	0,991027	0,996558
DT	0,997725	0,998994	0,998595	0,999483
RF	0,999912	0,999666	0,999723	0,999607
QDA	0,999694	0,998668	0,998675	0,999077
LDA	0,996123	0,995617	0,993863	0,996866
AB	0,996969	0,996391	0,995478	0,995957
NN	0,998952	0,997719	0,998401	0,996574
NB	0,99264	0,961234	0,98203	0,992231
GB	0,995496	0,998541	0,998985	0,998684



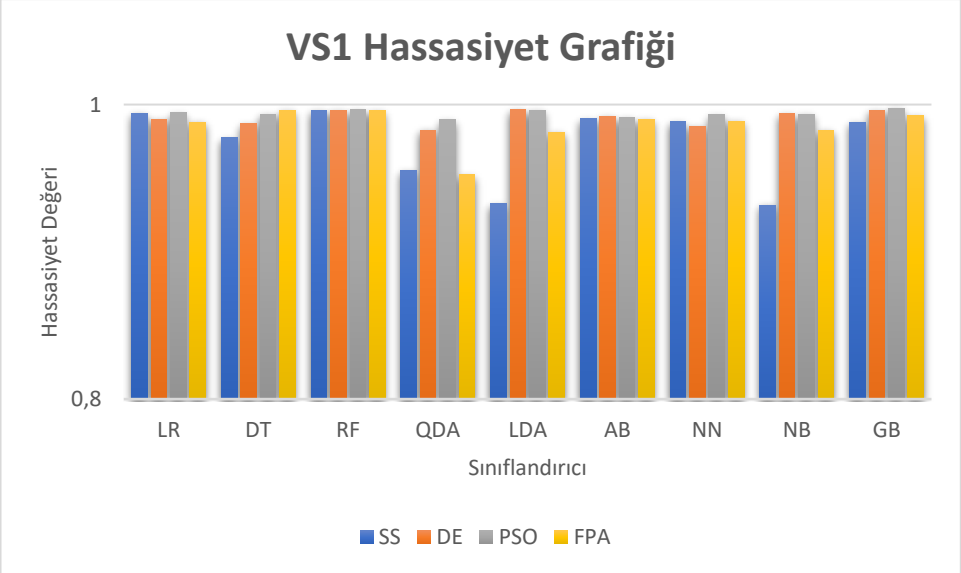
Şekil 5.4: VS1 Kesinlik-Sınıflandırıcı Sonuçları.

Kesinlik ölçütü açısından SS ile özellik seçimi yaklaşımları kullanılarak yapılan sınıflandırma sonuçları karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT ve GB'de, FPA algoritmasının LDA'da daha iyi sonuçlar elde ettiği görülmektedir.

Hassasiyet ölçütü için analiz sonuçları; tablo 5.4 ve şekil 5.5'te verilmiştir.

Tablo 5.4: VS1 Hassasiyet Sonuçları.

VS1-Hassasiyet				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,993972	0,990084	0,994344	0,988156
DT	0,977361	0,986861	0,993232	0,996323
RF	0,996279	0,996083	0,99649	0,995724
QDA	0,954927	0,982551	0,989772	0,952422
LDA	0,932575	0,996433	0,99609	0,981331
AB	0,990459	0,991987	0,991206	0,990204
NN	0,988411	0,985399	0,993024	0,98855
NB	0,93162	0,993739	0,993311	0,982639
GB	0,98803	0,996118	0,997429	0,992463



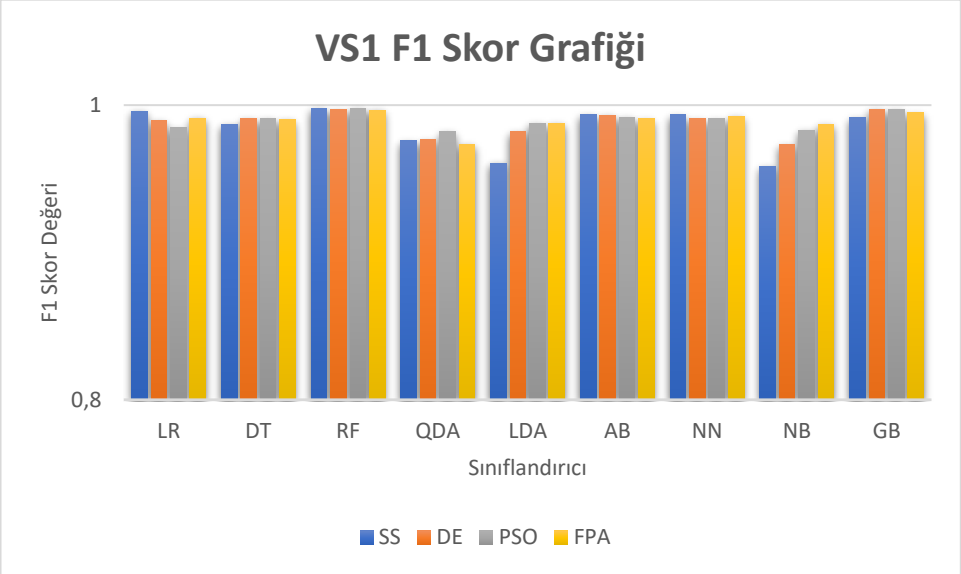
Şekil 5.5: VS1 Hassasiyet-Sınıflandırıcı Sonuçları.

Hassasiyet ölçütü açısından SS ile özellik seçimi yaklaşımları kullanılarak yapılan sınıflandırma sonuçları karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, QDA, AB, NB ve GB'de, PSO algoritmasının LR'de, PSO ve FPA algoritmalarının RF, NN'de, DE ve FPA algoritmalarının LDA'da daha iyi sonuçlar elde ettiği görülmektedir.

F1 Skor ölçütü için analiz sonuçları; tablo 5.5 ve şekil 5.6'da verilmiştir.

Tablo 5.5: VS1 F1 Skor Sonuçları.

VS1-F1 Skor				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,995946	0,989409	0,984548	0,991209
DT	0,987213	0,990728	0,991027	0,990164
RF	0,998078	0,996733	0,997864	0,996636
QDA	0,975932	0,976498	0,981859	0,973244
LDA	0,960511	0,98203	0,987668	0,987838
AB	0,993612	0,993231	0,991781	0,990919
NN	0,993514	0,990835	0,990838	0,992404
NB	0,958481	0,973215	0,983055	0,987204
GB	0,991684	0,997205	0,996864	0,994769

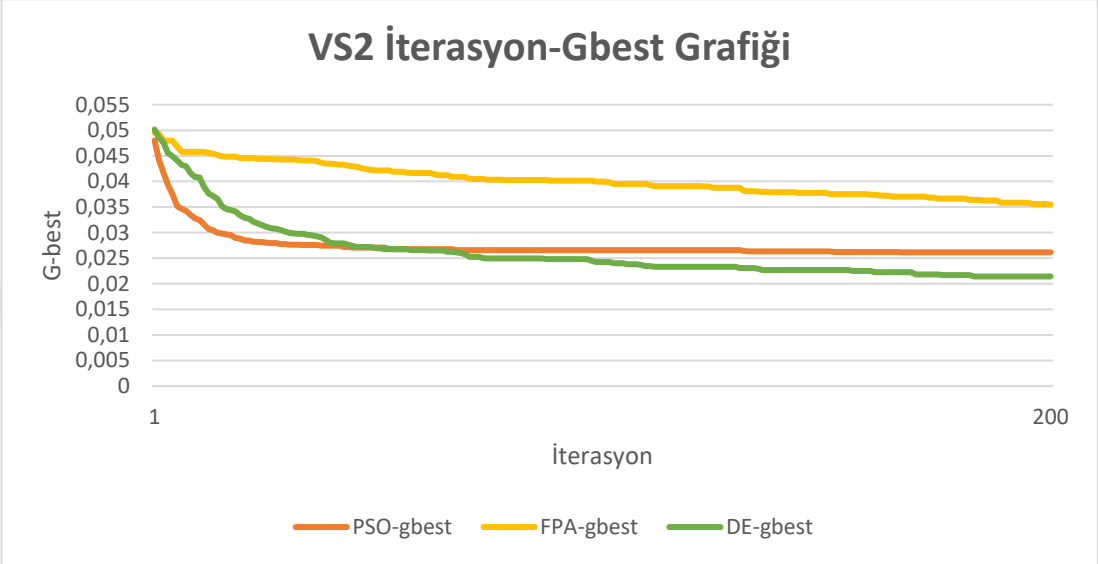


Şekil 5.6: VS1 F1 Skor-Sınıflandırıcı Sonuçları.

F1 Skor ölçütü açısından SS ile özellik seçimi yaklaşımları kullanılarak yapılan sınıflandırma sonuçları karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, LDA, NB, GB'de, PSO ve FPA algoritmalarının RF'de, DE ve PSO algoritmalarının QDA'da, PSO algoritmasının AB ve NN'de daha iyi sonuçlar elde ettiği görülmektedir.

5.2 NSL-KDD ile İlgili Çalışmalar

VS2 olarak ifade edilen NSL-KDD veri seti için yapılan çalışmalar; süre, doğruluk, kesinlik, hassasiyet ve F1 skor ölçütleri kullanılarak analiz edilmiştir. Bunlar aşağıda sırasıyla tablo ve şekillerde verilmiştir. PSO, FPA ve DE algoritmalarının 200 iterasyon için gbest değerleri aşağıda bulunan şekil 5.7'de verilmiştir.



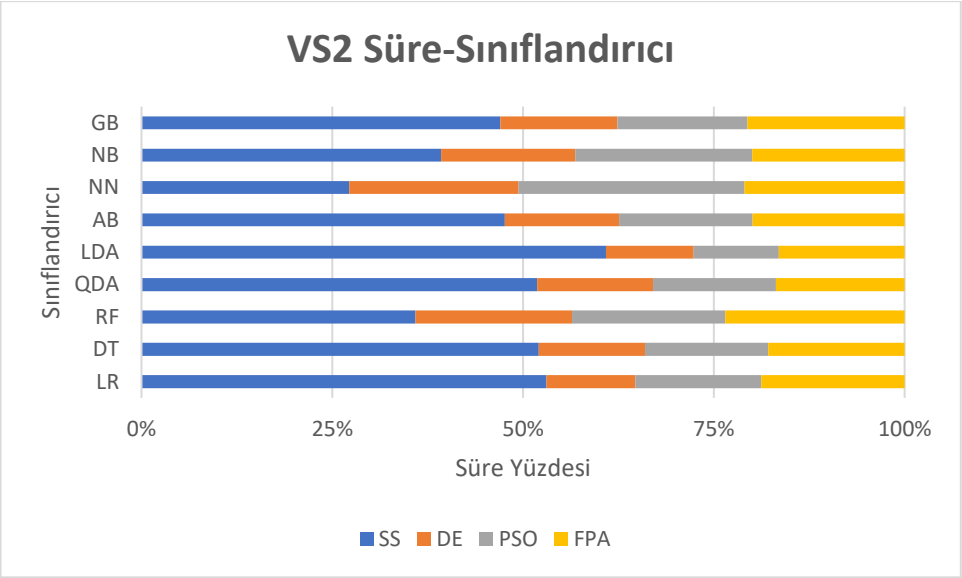
Şekil 5.7: VS2 İterasyon-Gbest Sonuçları.

Şekil 5.7'de görüldüğü üzere, yaklaşık 170. iterasyondan sonra FPA, PSO ve DE için en iyi gbest değeri elde edilmiştir.

Süre için analiz sonuçları; tablo 5.6 ve şekil 5.8'de verilmiştir.

Tablo 5.6: VS2 Süre Sonuçları.

VS2-Süre				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,707589	0,155747	0,21951	0,250832
DT	0,37702	0,10065	0,117091	0,129233
RF	3,966768	2,261828	2,222689	2,591954
QDA	0,271745	0,079538	0,084328	0,088415
LDA	0,659943	0,125035	0,120574	0,179003
AB	4,47684	1,41079	1,645681	1,872332
NN	20,69413	16,85782	22,49588	15,92133
NB	0,171541	0,076635	0,101432	0,087136
GB	6,5403	2,130754	2,371657	2,863361



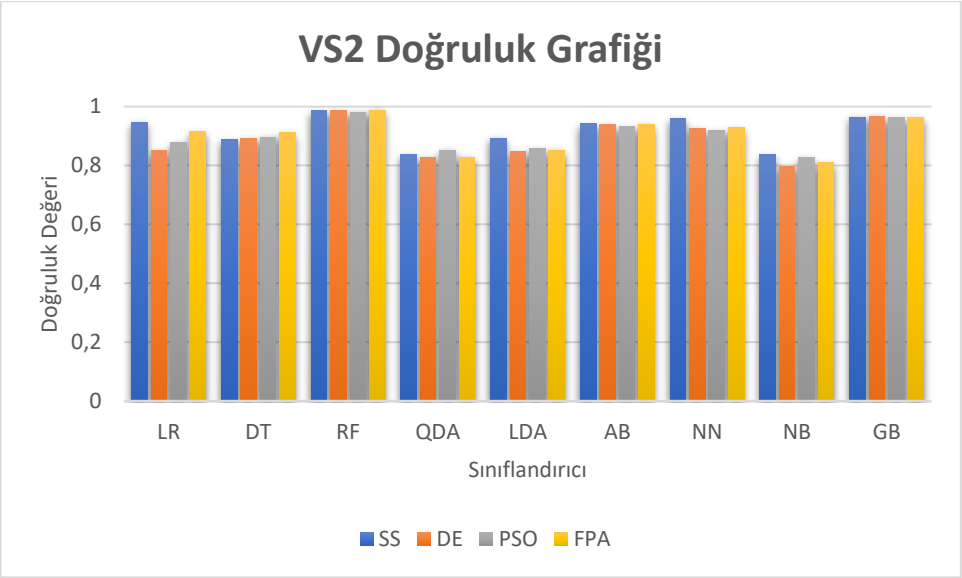
Şekil 5.8: VS2 Süre-Sınıflandırıcı Sonuçları.

Tablo 5.6 ve şekil 5.8 incelendiğinde, SS sonuçlarına göre özellik seçimi yaklaşımları kullanıldığında sınıflandırma süreleri 3'te 1 oranına gerileyerek süre açısından performansının arttığı görülmüştür.

Doğruluk ölçütü için analiz sonuçları; tablo 5.7 ve şekil 5.9'da verilmiştir.

Tablo 5.7: VS2 Doğruluk Sonuçları.

VS2-Doğruluk				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,946017	0,849938	0,878238	0,914301
DT	0,886976	0,891235	0,895759	0,910753
RF	0,986382	0,986116	0,981325	0,98789
QDA	0,836941	0,827981	0,852156	0,828735
LDA	0,89332	0,846123	0,857567	0,852245
AB	0,941669	0,938298	0,932221	0,940827
NN	0,9599	0,925701	0,917628	0,929382
NB	0,836276	0,795157	0,827183	0,809173
GB	0,964115	0,967131	0,963006	0,962695



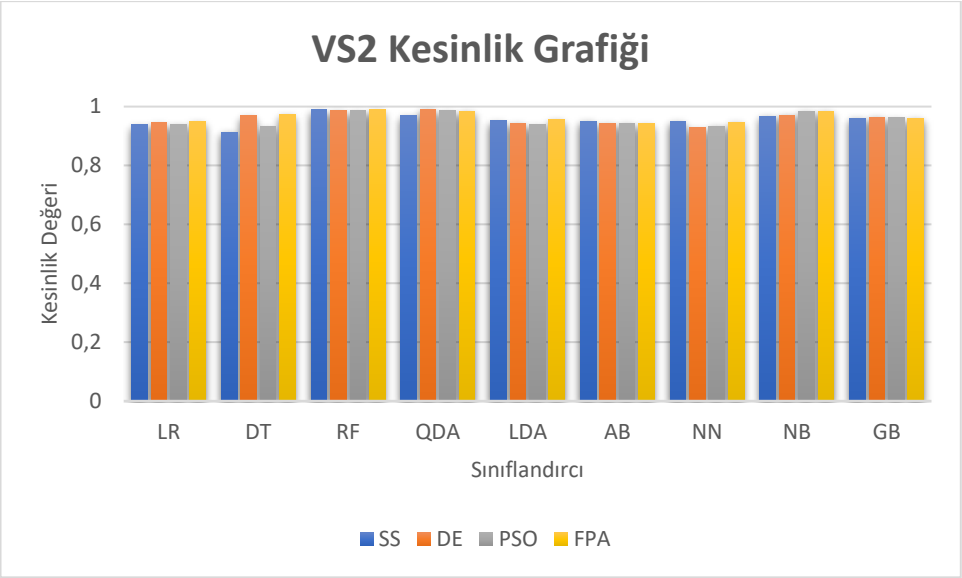
Şekil 5.9: VS2 Doğruluk-Sınıflandırıcı Sonuçları.

Doğruluk ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT'de, FPA algoritmasının RF'de, PSO algoritmasının QDA'da, DE algoritmasının GB'de daha iyi sonuçlar elde ettiği görülmüştür.

Kesinlik ölçütü için analiz sonuçları; tablo 5.8 ve şekil 5.10'da verilmiştir.

Tablo 5.8: VS2 Kesinlik Sonuçları.

VS2-Kesinlik				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,938547	0,945341	0,937994	0,947979
DT	0,913181	0,968827	0,93183	0,972616
RF	0,988613	0,986028	0,986589	0,989484
QDA	0,969521	0,990845	0,985304	0,984425
LDA	0,951843	0,944241	0,938877	0,955367
AB	0,949927	0,94282	0,944317	0,941943
NN	0,950336	0,927453	0,933536	0,946165
NB	0,967327	0,970836	0,983198	0,984244
GB	0,960655	0,964008	0,96135	0,960388



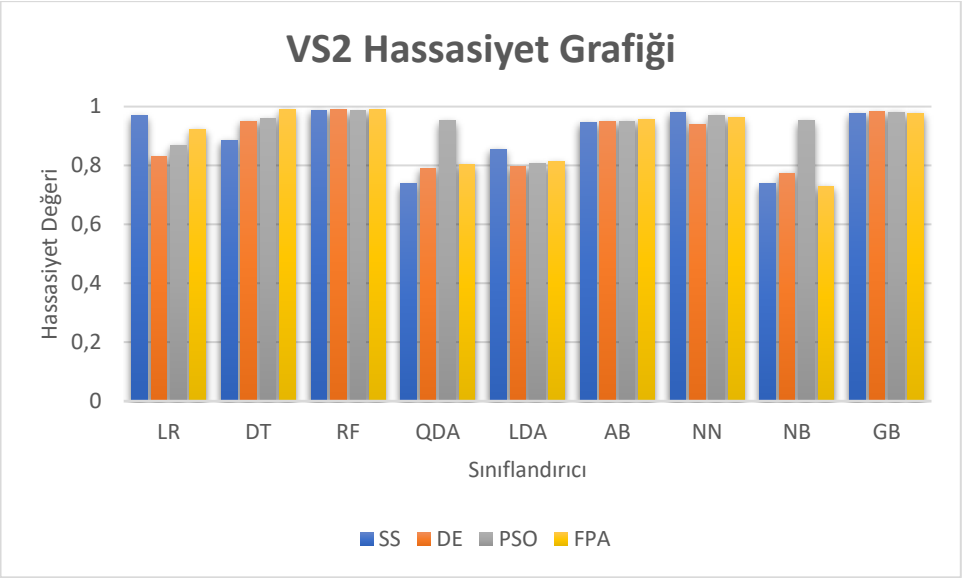
Şekil 5.10: VS2 Kesinlik-Sınıflandırıcı Sonuçları.

Kesinlik ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, QDA, NB'de, DE ve FPA algoritmalarının LR'de, DE ve PSO algoritmalarının GB'de, FPA algoritmasının RF, LDA'da daha iyi sonuçlar elde ettiği görülmüştür.

Hassasiyet ölçütü için analiz sonuçları; tablo 5.9 ve şekil 5.11'de verilmiştir.

Tablo 5.9: VS2 Hassasiyet Sonuçları.

VS2-Hassasiyet				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,968597	0,829815	0,868777	0,922076
DT	0,885683	0,949665	0,958933	0,990259
RF	0,987454	0,989636	0,986285	0,990026
QDA	0,73756	0,790696	0,952081	0,803876
LDA	0,855919	0,796932	0,80628	0,812749
AB	0,947478	0,950907	0,950129	0,955272
NN	0,980831	0,939063	0,969999	0,96213
NB	0,73732	0,773784	0,952393	0,728603
GB	0,977012	0,982545	0,981376	0,97787



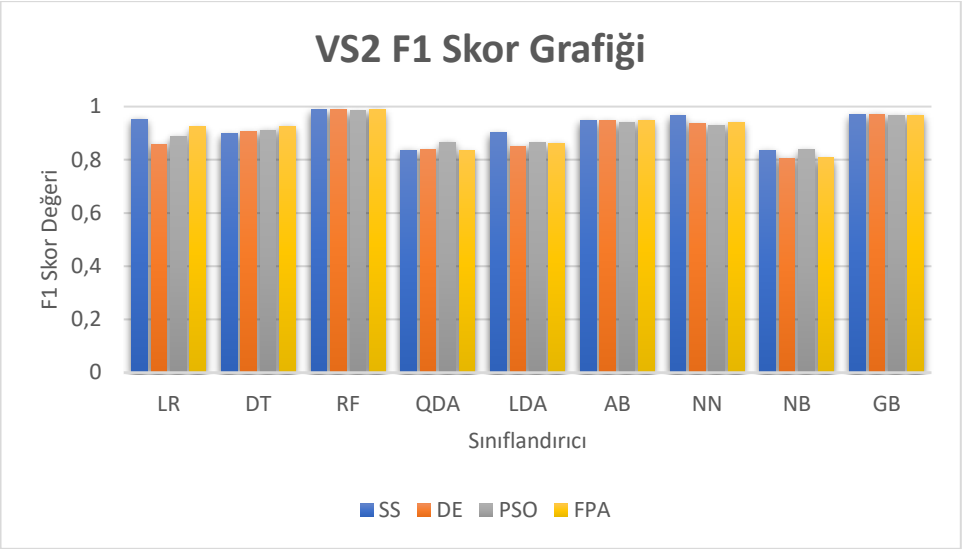
Şekil 5.11: VS2 Hassasiyet-Sınıflandırıcı Sonuçları.

Hassasiyet ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, QDA, AB, GB'de, DE ve FPA algoritmalarının RF'de, DE ve PSO algoritmalarının NB'de, DE algoritmasının LDA'da daha iyi sonuçlar elde ettiği görülmüştür.

F1 Skor ölçütü için analiz sonuçları; tablo 5.10 ve şekil 5.12'de verilmiştir.

Tablo 5.10: VS2 F1 Skor Sonuçları.

VS2-F1 Skor				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,953324	0,855696	0,887247	0,924525
DT	0,898903	0,906034	0,911167	0,926652
RF	0,988032	0,987828	0,983641	0,989363
QDA	0,836428	0,838082	0,866577	0,833296
LDA	0,901325	0,85155	0,865036	0,862281
AB	0,948673	0,94607	0,940648	0,948351
NN	0,965335	0,935293	0,930592	0,938426
NB	0,836758	0,806099	0,837743	0,807162
GB	0,968747	0,971346	0,967857	0,967475

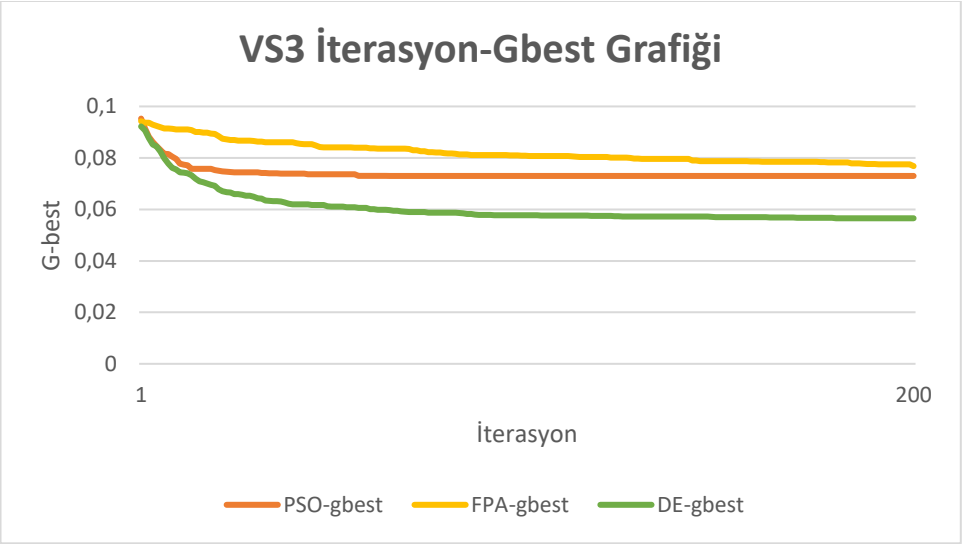


Şekil 5.12: VS2 F1 Skor-Sınıflandırıcı Sonuçları.

F1 Skor ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT'de, DE ve PSO algoritmalarının QDA'da, FPA algoritmasının RF'de, PSO algoritmasının NB'de, DE algoritmasının GB'de daha iyi sonuçlar elde ettiği görülmüştür.

5.3 UNSW-NB15 ile İlgili Çalışmalar

VS3 olarak ifade edilen UNSW-NB15 veri seti için yapılan çalışmalar; süre, doğruluk, kesinlik, hassasiyet ve F1 skor ölçütleri kullanılarak analiz edilmiştir. Bunlar aşağıda sırasıyla tablo ve şekillerde verilmiştir. PSO, FPA ve DE algoritmalarının 200 iterasyon için gbest değerleri aşağıda bulunan şekil 5.13'te verilmiştir.



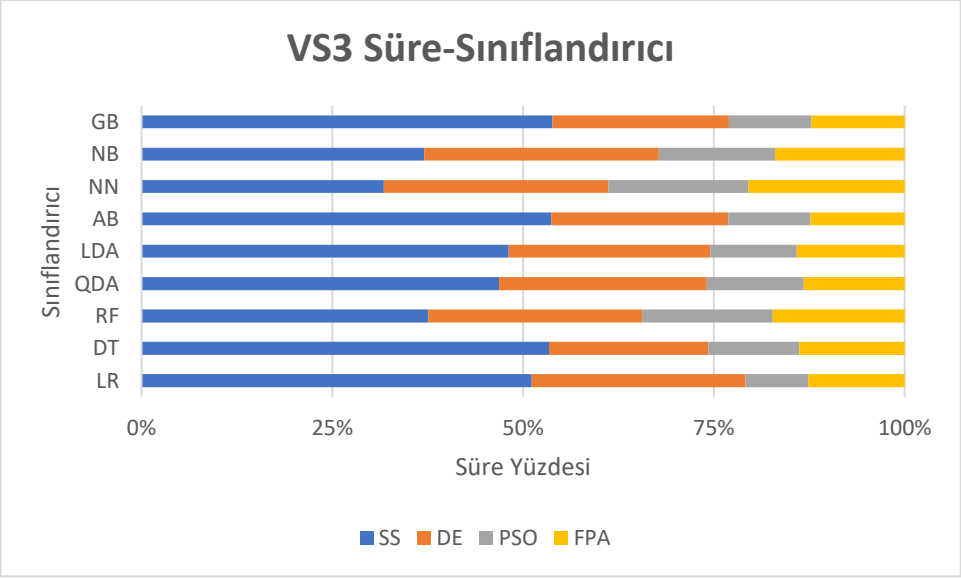
Şekil 5.13: VS3 İterasyon-Gbest Sonuçları.

Şekil 5.13'te görüldüğü üzere, yaklaşık 175. iterasyondan sonra FPA, PSO ve DE için en iyi gbest değeri elde edilmiştir.

Süre için analiz sonuçları; tablo 5.11 ve şekil 5.14'te verilmiştir.

Tablo 5.11: VS3 Süre Sonuçları.

VS3-Süre				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	3,155	1,7266	0,5069	0,7797
DT	2,205	0,8624	0,4919	0,57
RF	45,039	33,5754	20,5083	20,7078
QDA	0,839	0,4872	0,2272	0,2366
LDA	1,965	1,0799	0,4636	0,577
AB	46,133	19,9514	9,2616	10,6067
NN	71,019	65,8602	40,9688	45,7891
NB	0,51	0,4229	0,2096	0,2337
GB	79,9	34,3057	16,0578	18,111



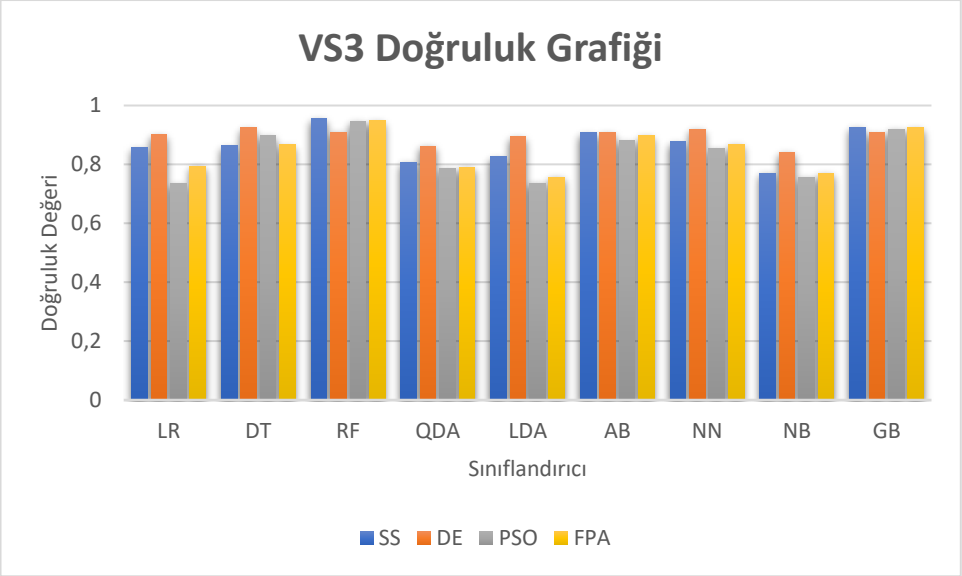
Şekil 5.14: VS3 Süre-Sınıflandırıcı Sonuçları.

Yukarıda verilen tablo ve grafik incelendiğinde; SS sonuçlarına göre özellik seçimi yaklaşımlarından DE kullanıldığında sınıflandırma süreleri 3'te 1 oranına gerilediği, PSO ve FPA kullanıldığında 4'te 1 oranına gerileyerek süre açısından performansının arttığı görülmüştür.

Doğruluk ölçütü için analiz sonuçları; tablo 5.12 ve şekil 5.15'te verilmiştir.

Tablo 5.12: VS3 Doğruluk Sonuçları.

VS3-Doğruluk				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,856628	0,900211	0,734559	0,793212
DT	0,863297	0,924438	0,896758	0,867061
RF	0,954525	0,90977	0,94708	0,948562
QDA	0,808175	0,861806	0,787833	0,791061
LDA	0,826663	0,893499	0,736903	0,756641
AB	0,908358	0,907825	0,87968	0,89671
NN	0,876378	0,920286	0,853094	0,869101
NB	0,770562	0,841891	0,757285	0,76891
GB	0,926832	0,907289	0,919824	0,924026



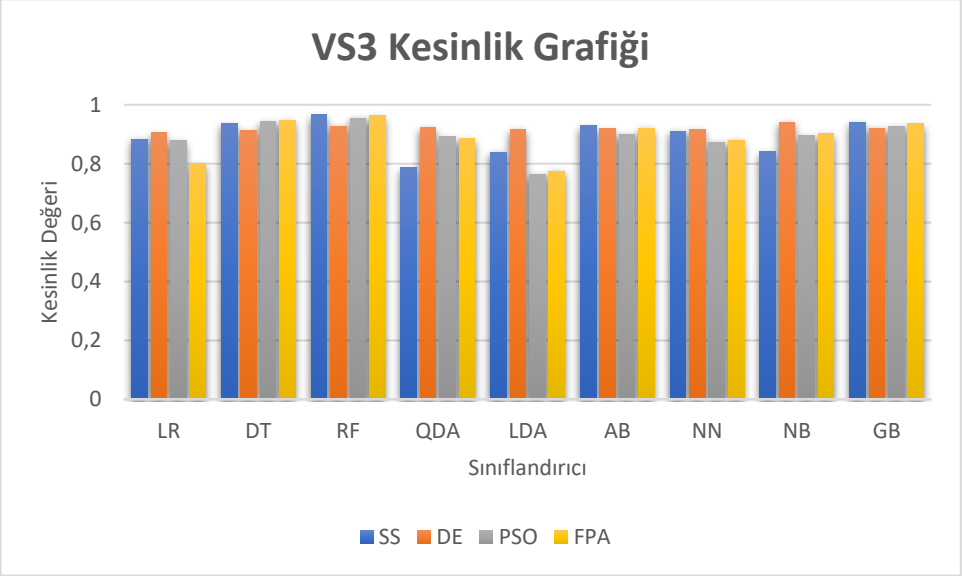
Şekil 5.15: VS3 Doğruluk-Sınıflandırıcı Sonuçları.

Doğruluk ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT'de, DE algoritmasının LR, QDA, LDA, NN, NB'de daha iyi sonuçlar elde ettiği görülmüştür.

Kesinlik ölçütü için analiz sonuçları; tablo 5.13 ve şekil 5.16'de verilmiştir.

Tablo 5.13: VS3 Kesinlik Sonuçları.

VS3-Kesinlik				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,883328	0,90424	0,877871	0,800752
DT	0,936154	0,913014	0,944603	0,946704
RF	0,966563	0,927162	0,953704	0,963338
QDA	0,785952	0,924052	0,891149	0,885217
LDA	0,838788	0,917377	0,763985	0,774519
AB	0,928105	0,918563	0,89927	0,920234
NN	0,91033	0,916308	0,87241	0,879882
NB	0,839912	0,940723	0,894108	0,901909
GB	0,941363	0,920381	0,927342	0,934855



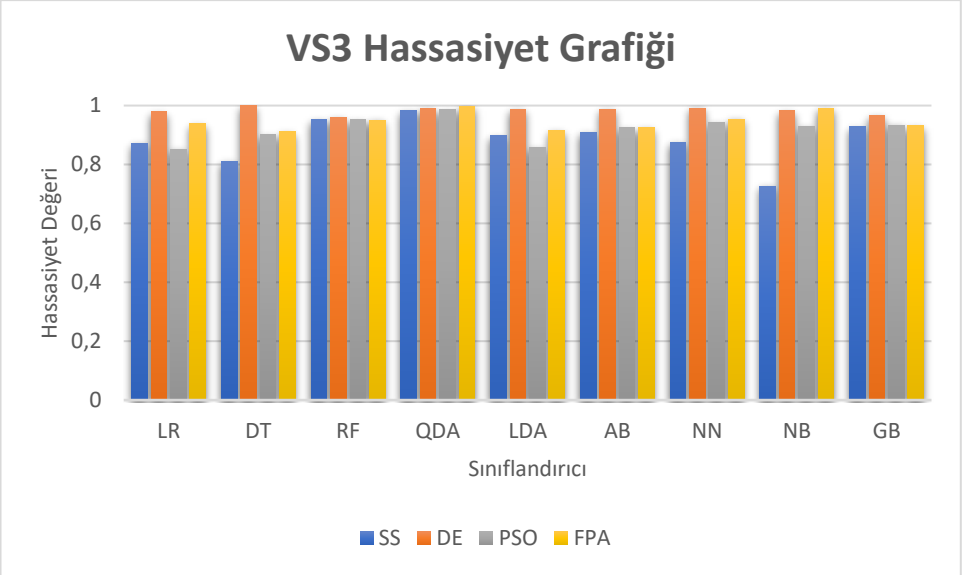
Şekil 5.16: VS3 Kesinlik-Sınıflandırıcı Sonuçları.

Kesinlik ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının QDA'da, PSO ve FPA algoritmalarının DT'de, DE ve PSO algoritmalarının NB'de, DE algoritmasının LR, LDA, NN'de daha iyi sonuçlar elde ettiği görülmüştür.

Hassasiyet ölçütü için analiz sonuçları; tablo 5.14 ve şekil 5.17'de verilmiştir.

Tablo 5.14: VS3 Hassasiyet Sonuçları.

VS3-Hassasiyet				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,872212	0,979772	0,851541	0,938171
DT	0,810334	0,999784	0,900512	0,91152
RF	0,951955	0,958288	0,952705	0,949484
QDA	0,982574	0,99157	0,986235	0,997353
LDA	0,898087	0,987255	0,856047	0,915208
AB	0,91002	0,987104	0,925594	0,923917
NN	0,875302	0,990875	0,94108	0,954272
NB	0,725888	0,982822	0,928066	0,990735
GB	0,92855	0,965896	0,933469	0,931726



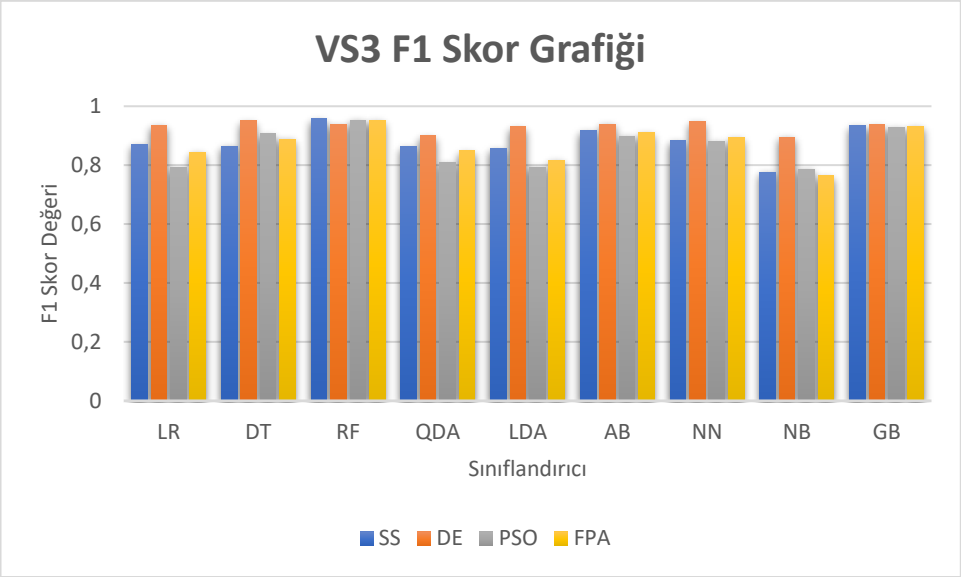
Şekil 5.17: VS3 Hassasiyet-Sınıflandırıcı Sonuçları.

Hassasiyet ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, QDA, AB, NN, NB, GB'de, DE ve FPA algoritmalarının LR, LDA'da, DE ve PSO algoritmalarının RF'de daha iyi sonuçlar elde ettiği görülmüştür.

F1 Skor ölçütü için analiz sonuçları; tablo 5.15 ve şekil 5.18'de verilmiştir.

Tablo 5.15: VS3 F1 Skor Sonuçları.

VS3-F1 Skor				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,869962	0,935062	0,791083	0,842999
DT	0,864012	0,950656	0,906819	0,887857
RF	0,958257	0,937903	0,952105	0,952395
QDA	0,862075	0,900705	0,808619	0,848316
LDA	0,856551	0,931026	0,793587	0,816571
AB	0,916241	0,938919	0,897489	0,909432
NN	0,882217	0,947569	0,881151	0,892414
NB	0,776166	0,895556	0,785929	0,764731
GB	0,933347	0,937176	0,928882	0,931848

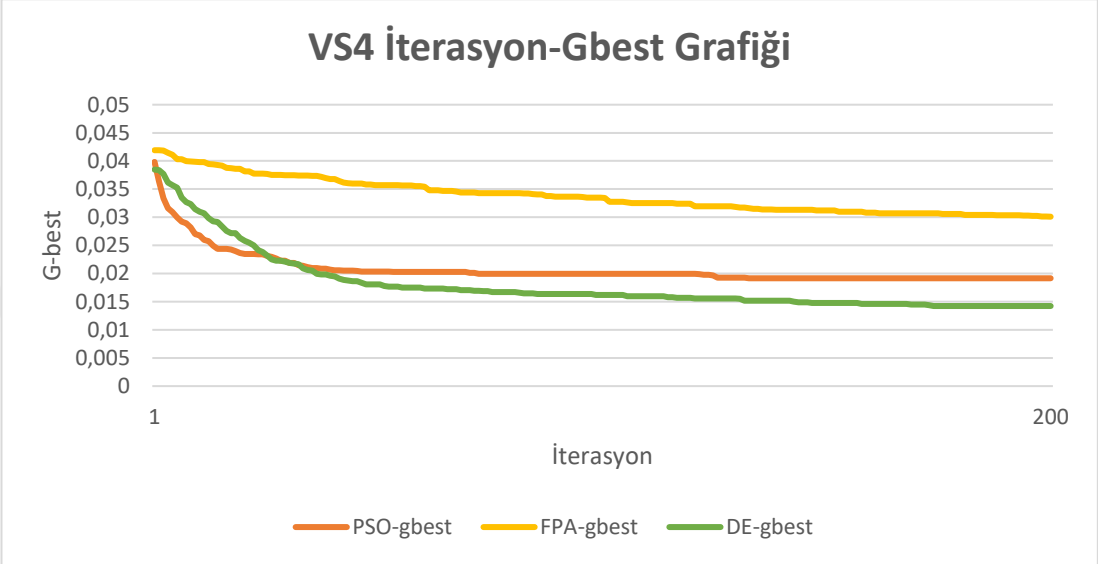


Şekil 5.18: VS3 F1 Skor-Sınıflandırıcı Sonuçları.

F1 Skor ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT'de, DE ve FPA algoritmalarının NN'de, DE ve PSO algoritmalarının NB'de, DE algoritmasının LR, QDA, LDA, AB, GB'de daha iyi sonuçlar elde ettiği görülmüştür.

5.4 CSE-CIC-IDS2018 ile İlgili Çalışmalar

VS4 olarak ifade edilen CSE-CIC-IDS2018 veri seti için yapılan çalışmalar; süre, doğruluk, kesinlik, hassasiyet ve F1 skor ölçütleri kullanılarak analiz edilmiştir. Bunlar aşağıda sırasıyla tablo ve şekillerde verilmiştir. PSO, FPA ve DE algoritmalarının 200 iterasyon için gbest değerleri aşağıda bulunan şekil 5.19'de verilmiştir.



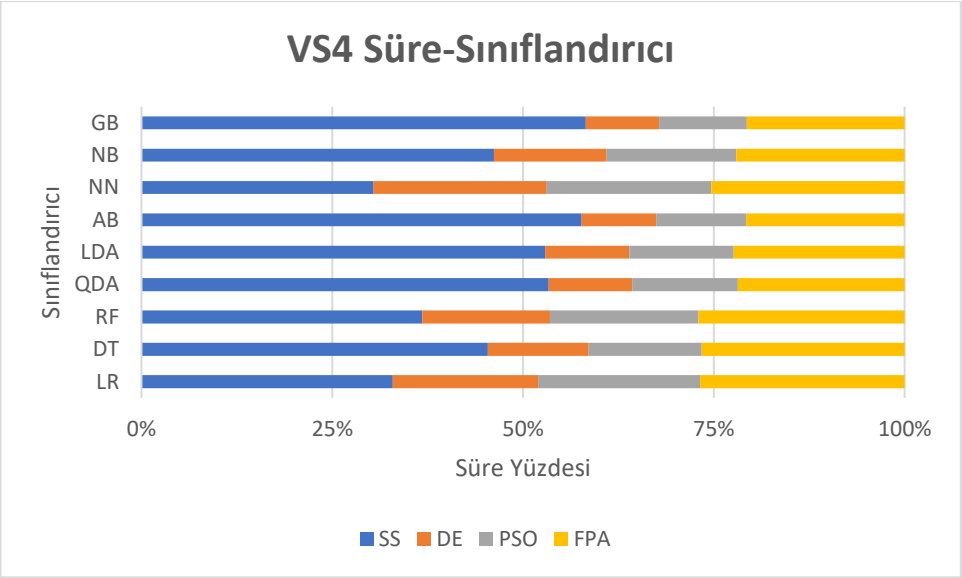
Şekil 5.19: VS4 İterasyon-Gbest Sonuçları.

Şekil 5.19'da görüldüğü üzere, yaklaşık 160. iterasyondan sonra FPA, PSO ve DE için en iyi gbest değeri elde edilmiştir

Süre için analiz sonuçları; tablo 5.16 ve şekil 5.20'de verilmiştir.

Tablo 5.16: VS4 Süre Sonuçları.

VS4-Süre				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	26,395	15,2811	17,0172	21,4997
DT	9,693	2,8282	3,1623	5,6807
RF	97,441	44,321	51,5203	71,4856
QDA	10,471	2,15	2,7112	4,2903
LDA	13,564	2,8238	3,5062	5,7416
AB	432,353	73,2679	88,3886	155,4951
NN	200,291	149,6252	141,9048	167,0489
NB	4,5	1,4427	1,6543	2,1501
GB	758,365	125,4525	149,9201	269,2167



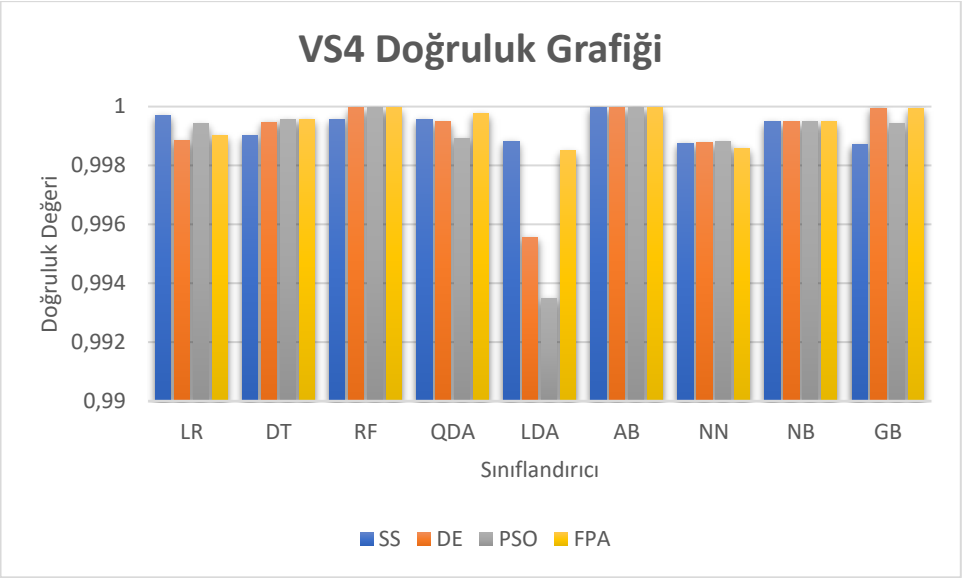
Şekil 5.20: VS4 Süre-Sınıflandırıcı Sonuçları.

Yukarıda verilen tablo ve grafik incelendiğinde, SS sonuçlarına göre özellik seçimi yaklaşımları kullanıldığında sınıflandırma süreleri 6'da 1 oranına kadar gerileyerek süre açısından performansının arttığı görülmüştür.

Doğruluk ölçütü için analiz sonuçları; tablo 5.17 ve şekil 5.21'de verilmiştir.

Tablo 5.17: VS4 Doğruluk Sonuçları.

VS4-Doğruluk				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,999699	0,998848	0,999418	0,999002
DT	0,999021	0,999476	0,999557	0,999568
RF	0,999568	0,999964	0,99997	0,999965
QDA	0,999573	0,999484	0,998919	0,999758
LDA	0,998831	0,995562	0,993478	0,998503
AB	0,999983	0,999956	0,999983	0,999954
NN	0,998762	0,998773	0,998807	0,998583
NB	0,999487	0,999484	0,999487	0,999487
GB	0,998723	0,999944	0,999419	0,999937



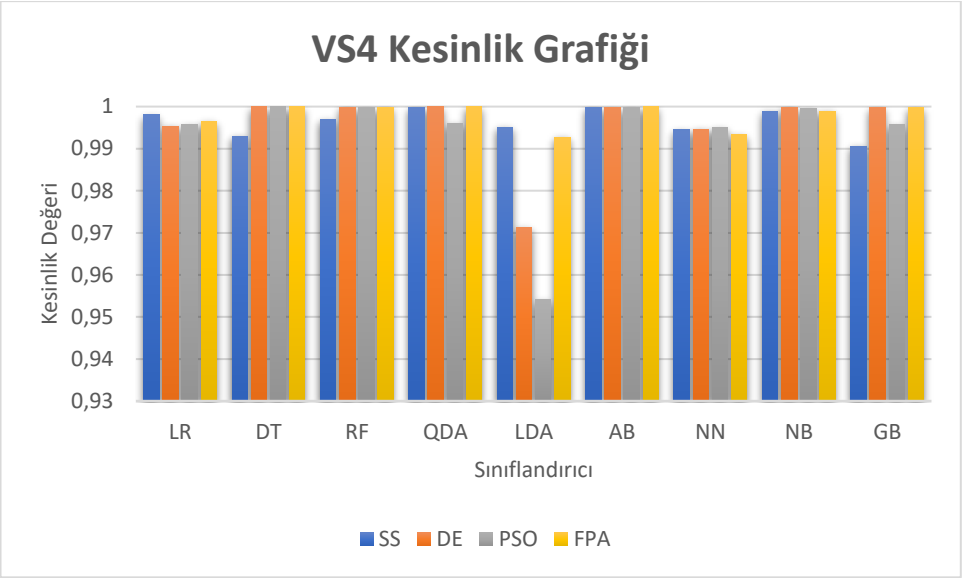
Şekil 5.21: VS4 Doğruluk-Sınıflandırıcı Sonuçları.

Doğruluk ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, GB'de, DE ve FPA algoritmalarının RF'de, DE ve PSO algoritmalarının NN'de, FPA algoritmasının QDA'da daha iyi sonuçlar elde ettiği görülmüştür.

Kesinlik ölçütü için analiz sonuçları; tablo 5.18 ve şekil 5.22'de verilmiştir.

Tablo 5.18: VS4 Kesinlik Sonuçları.

VS4-Kesinlik				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,998065	0,99537	0,995739	0,996459
DT	0,992917	0,999976	0,999988	0,999953
RF	0,996919	0,999835	0,999882	0,999894
QDA	0,999897	0,999976	0,995987	0,999988
LDA	0,995137	0,971344	0,954101	0,992768
AB	0,999894	0,999693	0,999882	0,999988
NN	0,994463	0,994581	0,994961	0,993289
NB	0,998947	0,999865	0,999539	0,998867
GB	0,99058	0,99967	0,99573	0,99967



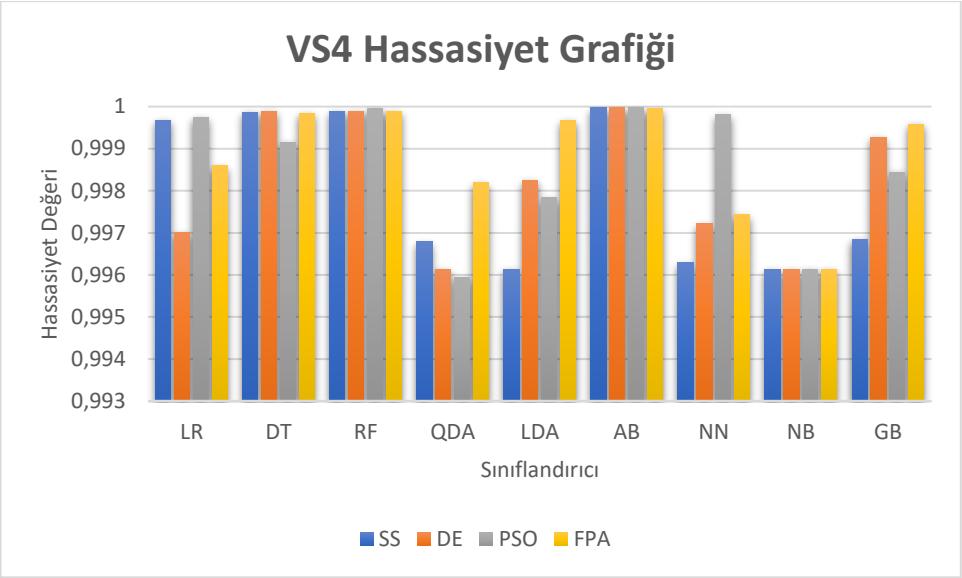
Şekil 5.22: VS4 Kesinlik-Sınıflandırıcı Sonuçları.

Kesinlik ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, RF, GB'de, DE ve FPA algoritmalarının QDA'da, DE ve PSO algoritmalarının NN, NB'de, FPA algoritmasının AB'de daha iyi sonuçlar elde ettiği görülmüştür.

Hassasiyet ölçütü için analiz sonuçları; tablo 5.19 ve şekil 5.23'te verilmiştir.

Tablo 5.19: VS4 Hassasiyet Sonuçları.

VS4-Hassasiyet				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,999681	0,997001	0,999752	0,998595
DT	0,99987	0,999894	0,99915	0,999835
RF	0,999882	0,999894	0,999965	0,999882
QDA	0,996788	0,996139	0,995938	0,998193
LDA	0,996139	0,998247	0,997836	0,999669
AB	0,999976	0,999978	0,999988	0,999953
NN	0,996292	0,997213	0,999811	0,997426
NB	0,996139	0,996139	0,996139	0,996139
GB	0,996845	0,999267	0,998439	0,999583



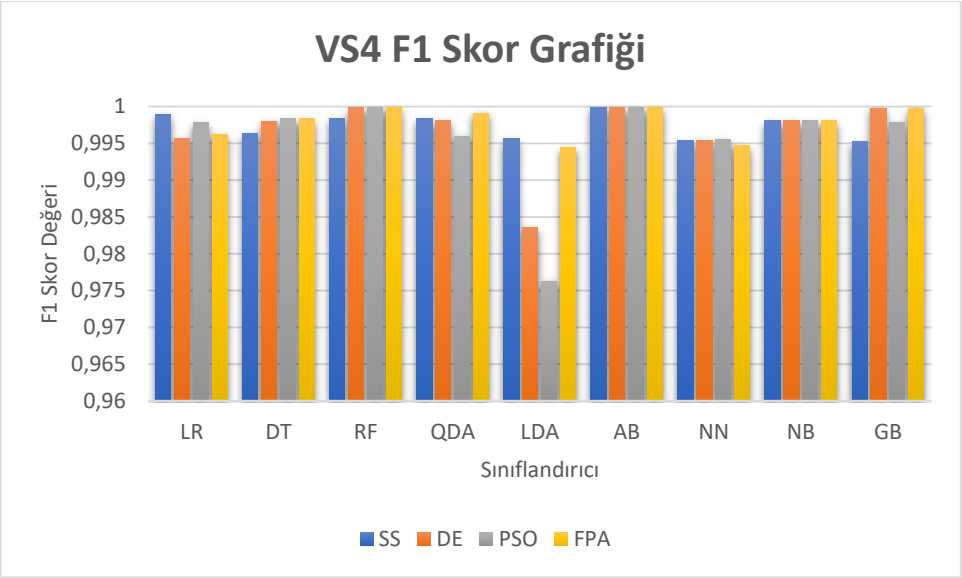
Şekil 5.23: VS4 Hassasiyet-Sınıflandırıcı Sonuçları.

Hassasiyet ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının LDA, NN, GB'de, DE ve PSO algoritmalarının RF, AB'de, PSO algoritmasının LR'de, DE algoritmasının DT'de, FPA algoritmasının QDA'da daha iyi sonuçlar elde ettiği görülmüştür.

F1 Skor ölçütü için analiz sonuçları; tablo 5.20 ve şekil 5.24'te verilmiştir.

Tablo 5.20: VS4 F1 Skor Sonuçları.

VS4-F1 Skor				
Özellik Seçimi / Sınıflandırıcılar	SS	DE	PSO	FPA
LR	0,99887	0,995686	0,997815	0,996245
DT	0,996355	0,998033	0,998336	0,998378
RF	0,998389	0,999864	0,999888	0,99987
QDA	0,998391	0,998054	0,995948	0,999089
LDA	0,995622	0,983524	0,97627	0,994391
AB	0,999935	0,999835	0,999935	0,999829
NN	0,995363	0,995404	0,995534	0,994692
NB	0,998066	0,998054	0,998066	0,998066
GB	0,995243	0,999788	0,997837	0,999764



Şekil 5.24: VS4 F1 Skor-Sınıflandırıcı Sonuçları.

F1 Skor ölçütü açısından SS ile özellik seçimi uygulanan sınıflandırıcılar karşılaştırıldığında; DE, PSO ve FPA algoritmalarının DT, RF, GB'de, De ve PSO algoritmalarının NN'de, FPA algoritmasının QDA'da daha iyi sonuçlar elde ettiği görülmüştür.

6. SONUÇLAR VE ÖNERİLER

Günümüzde teknolojinin ilerlemesiyle birlikte internete bağlı cihaz ve internet kullanan birey sayısı önemli ölçüde artmaktadır. Yakın geçmişte görüldüğü üzere, gelecekte de bu kullanımın önemli derecede artacağı öngörülmektedir. Dolayısıyla kişisel, ticari, gizli ve hassas veriler internette dolaşacaktır. Teknolojinin gelişmesi ve internet kullanımının artmasıyla birlikte, siber saldırı türleri de artarak gelişmektedir. Saldırganlar, sistemlerin çalışmasını etkileyebilecek, verilerin kullanılabilirliğini ve gizliliğini değiştirebilecek hassas verilere erişim sağlamak için güvenlik önlemlerindeki kusurlardan faydalanarak siber saldırı faaliyetlerini gerçekleştirirler. Siber güvenlik, siber saldırılarla mücadelede ve buna bağlı maliyet ve zararların azaltılmasında önemli bir alan olarak görülmektedir. Siber güvenliğin temel bileşenleri olan; Gizlilik, Bütünlük, Erişilebilirlik prensipleri açısından bilgilerin koruması büyük önem arz etmektedir.

Siber güvenlik açısından güvenlik mimarisinin önemli bir bileşeni olan Saldırı Tespit Sistemleri, çeşitli izinsiz giriş türlerini tanıyan ve uyarı oluşturan bir araç görevi görmektedir. STS bir ağı izler ve ağ trafiği aracılığıyla şüpheli ve kötü amaçlı etkinlikleri veya politika ihlallerini belirler; bu da ağ yöneticilerinin mevcut tehditleri sürekli izlemesine olanak tanır. STS'lerin temel görevi, sistem güvenliğini tehlikeye atan davetsiz misafirleri veya kötü niyetli etkinlikleri önlemek için çeşitli işlem kombinasyonları uygulayarak bir sistemi korumaktır.

STS'ler, ağ trafiğinin meşru olduğundan ve kötü amaçlı olmadığından emin olmak ve kötü amaçlı ise uyarı oluşturmak için sürekli olarak geliştirilmeye devam etmektedir. Bu gelişimin farkına varmak için STS'lerin çeşitleri doğru analiz edilmelidir. STS'ler, ana bilgisayar tabanlı ve ağ tabanlı olarak sınıflandırılır. Ana bilgisayar tabanlı STS'ler, yalnızca şüpheli eylemleri ve güvenlik politikası ihlallerini tarayarak cihazdaki veri paketlerini izleme mantığı ile çalışır ve tek bir ana bilgisayara veya cihaza dağıtılır. Bu çalışma mantığı korunması gereken her ana bilgisayara bir STS kurma zorunluluğu getirir ve her ana bilgisayar için performans kaybı oluşturacak işlem süresi getirir. Ağ tabanlı STS'ler, bilgisayar ağının tamamına dağıtılarak kötü amaçlı etkinlikleri tespit etmek amacıyla trafiği izlemek, yakalamak ve analiz etmek

için kullanılır. STS'ler analiz yaklaşımı bakımından anormallik tabanlı veya kötüye kullanıma dayalı (imza tabanlı) iki temel analiz yaklaşımı kullanır. İmza tabanlı STS'ler, bilinen saldırıların imzalarını veya önceden tanımlanmış normal aktivite profilinden sapmaları güncel bir veri tabanında tutar ve ağ üzerinde bulunan veriler ile karşılaştırır; eşleşme olması durumunda uyarı oluşturur. İmza Tabanlı STS'ler veri tabanındaki saldırıları tespit edebildiğinden bilinen saldırıları çok düşük yanlış pozitiflerle kolayca tanımlarken, yeni/sıfır gün saldırı türlerini tespit etme yeteneğinden yoksundur. Anormallik tabanlı STS'ler, sistemin normal davranışının bir modelini oluşturarak, modeldeki düzensizliklere dayalı saldırıları tanımlamak için modelden sapan etkinlikleri arar. Anormallik tabanlı STS'ler, yeni ve bilinmeyen saldırıları yani yeni/sıfır gün saldırı türlerini tespit etme yeteneği ile öne çıkmaktadır. Ancak daha yüksek bir yanlış pozitif oranının ortaya çıkması anormallik tabanlı STS'lerin uygulanmasını kısıtlar. Ağdaki ve bilgisayarlardaki tehditlerin listesi sonsuzdur ve sürekli olarak gelişmektedir; bu nedenle, yeni/sıfır gün saldırı türleri her geçen gün artmaktadır. Anormallik tabanlı STS'ler aktif bir araştırma alanı olmaya devam etmekte ve izinsiz girişleri tanımlamak için makine öğrenmesi tekniklerini yani yapay zekayı kullanan anormallik tabanlı yaklaşımlara odaklanılmaktadır.

Makine öğrenmesi yaklaşımları veri setleri üzerinde modellerin eğitim ve test aşamalarında kullanılarak sınıflandırma sonuçları değerlendirilmektedir. Makine öğrenmesi algoritmaları, eğitim sürecinde verilen özellikler üzerinden modeli eğiterek, saldırı tespiti için doğru tahminde bulunulmasını sağlarlar. Literatürde bu amaçla oluşturulmuş ve geliştirilmiş veri setleri bulunmaktadır. Veri setleri üzerinde her bir ağ trafiği örneğinin birden çok özelliği bulunmaktadır. Günümüz araştırmalarında saldırı tespit sistemi modellerinin performansını artırmak amacıyla, saldırı tahmininde etkili olan özelliklerin seçimi için meta sezgisel algoritmaların kullanımı yaklaşımı benimsenmektedir.

Bu çalışmada literatürde açık kaynak olarak bulunan KDD CUP 99, NSL-KDD, UNSW-NB15 ve CICIDS2018 veri setleri kullanılarak, meta sezgisel yaklaşımlar olan FPA, PSO ve DE algoritmaları ile lojistik regresyon, karar ağacı, rastgele orman, lineer diskriminant analiz, kuadratik diskriminant analiz, adaboost, k en yakın komşu, saf bayes, gradyan artırma ve sinir ağları makine öğrenmesi algoritmaları kullanan hibrit bir analiz modeli önerilmiştir. Bu modelin sonuçları sadece makine öğrenmesi

algoritmalarını kullanılarak elde edilen sınıflandırma sonuçları ile karşılaştırılarak analiz edilmiştir.

Bu analizin sonuçları, bize sadece makine öğrenmesi algoritmaları kullanılarak yapılan sınıflandırmaya kıyasla, özellik seçimi yaklaşımının süre açısından STS modelinin performansını yaklaşık %200 oranında arttığını göstermiştir. Tüm algoritmalarda süre açısından performans artışı sağlanırken, saldırı tespit etme yetenekleri değerlendirme ölçütleri ile değerlendirildiğinde veri setlerinde bulunan eksikliklerin sınıflandırma sonucu ile doğru orantılı olduğunu göstermiştir. Analiz sonuçlarında, özellik seçimimi yaklaşımlarında başarılı sonuçlar elde edildiği gözlemlenmiştir. VS1'de; DE algoritması DT, QDA, LDA, NB, GB'de sırasıyla 0,985497, 0,964215, 0,971441, 0,95633, 0,995529, PSO algoritması DT, RF, QDA, LDA, GB'de sırasıyla 0,985514, 0,996597, 0,971896, 0,980596, 0,994985, FPA algoritması DT, LDA, NB, GB'de sırasıyla, 0,984062, 0,980887, 0,979798, 0,991675 en iyi doğruluk oranına ulaşmıştır. VS2'de; DE algoritması DT, GB'de sırasıyla 0,891235, 0,967131, PSO algoritması DT, QDA'da sırasıyla 0,895759, 0,852156, FPA algoritması DT, RF'de 0,910753, 0,98789 en iyi doğruluk oranına ulaşmıştır. VS3'te; DE algoritması LR, DT, QDA, LDA, NN, NB'de sırasıyla 0,900211, 0,924438, 0,861806, 0,893499, 0,920286, 0,841891, PSO algoritması DT'de 0,896758, FPA algoritması DT'de 0,867061 en iyi doğruluk oranına ulaşmıştır. VS4'te; DE algoritması DT, RF, NN, GB'de sırasıyla 0,999476, 0,999964, 0,998773, 0,999944, PSO algoritması DT, NN, GB'de sırasıyla 0,999557, 0,998807, 0,999419, FPA algoritması DT, RF, QDA, GB'de sırasıyla 0,999568, 0,999965, 0,999758, 0,999937 en iyi doğruluk oranına ulaşmıştır.

Bu çalışmanın sonuçlarından yola çıkılarak, özellik seçimi yaklaşımlarının gelecekte yapılacak çalışmalar üzerinde olumlu etkiler yapacağı öngörülmektedir. Birincil olarak, veri seti oluşturma süreçleri sonuçları ve STS'lerin başarılarını doğrudan etkilediği bilinmeli ve bu doğrultuda çalışmalara ağırlık verilmelidir. Gelecekte oluşturulacak veya canlı olarak izlenen ağ trafiğinde, örnek ağ trafiği sayısı ve trafiğin özellik sayısı çok daha fazla olacaktır. Özellik seçimi yaklaşımları sınıflandırma sonucunu etkileyen doğru özelliklerin seçiminde başarılı olarak özellik sayısının azaltılması anormallik tabanlı STS'lerin gelişimini pozitif yönde etkileyecektir. Buradan yola çıkılarak özellik seçimi yaklaşımları benimsenmelidir.

KAYNAKÇA

Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019). Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. *Electronics 2019, Vol. 8, Page 322*, *8*(3), 322. https://doi.org/10.3390/ELECTRONICS8030322

Abedin Disha, R., & Waheed, S. (2022). *Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique*. *5*, 1. https://doi.org/10.1186/s42400-021-00103-8

Adhi Tama, B., Comuzzi, M., & Rhee, K. (2019). *TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System*. https://doi.org/10.1109/ACCESS.2019.2928048

Ahmad, I., Abdullah, A. B., & Alghamdi, A. S. (2009). Application of artificial neural network in detection of DOS attacks. *SIN'09 - Proceedings of the 2nd International Conference on Security of Information and Networks*, 229–234. https://doi.org/10.1145/1626195.1626252

Ahmad, I., Haq, Q. E. U., Imran, M., Alassafi, M. O., & Alghamdi, R. A. (2022). An Efficient Network Intrusion Detection and Classification System. *Mathematics 2022, Vol. 10, Page 530*, *10*(3), 530. https://doi.org/10.3390/MATH10030530

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1). https://doi.org/10.1002/ETT.4150

Ahmed, H. A., Hameed, A., & Bawany, N. Z. (2022). Network intrusion detection using oversampling technique and machine learning algorithms. *PeerJ Computer Science*, *8*, e820. https://doi.org/10.7717/PEERJ-CS.820/FIG-2

Akasapu, S. (2017). *International Journal on Recent and Innovation Trends in Computing and Communication An Integrated Approach for detecting DDoS attacks in Cloud Computing*. Retrieved from http://www.ijritcc.org

Alazzam, H., Sharieh, A., & Eddin Sabri, K. (2021). *A lightweight intelligent network intrusion detection system using OCSVM and Pigeon inspired optimizer*. https://doi.org/10.1007/s10489-021-02621-x

Alazzam, H., Sharieh, A., & Sabri, K. E. (2020). A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer. *Expert Systems With Applications*, *148*, 113249. https://doi.org/10.1016/j.eswa.2020.113249

Aljawarneh, S., Aldwairi, M., & Yassein, B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, *25*, 152–160. https://doi.org/10.1016/j.jocs.2017.03.006

Anwar, S., Mohamad Zain, J., Fadli Zolkipli, M., Inayat, Z., Khan, S., Anthony, B., … Mi Mi Aung, K. (2017). *Algorithms Review From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions*. https://doi.org/10.3390/a10020039

Asharf, J., Moustafa, N., Khurshid, H., Debie, E., Haider, W., & Wahab, A. (2020). *electronics A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions*. https://doi.org/10.3390/electronics9071177

Ashiku, L., & Dagli, C. (2021). Network Intrusion Detection System using Deep Learning. *Procedia Computer Science*, *185*, 239–247. https://doi.org/10.1016/J.PROCS.2021.05.025

Boden, M. A. (1987). *Artificial Intelligence and Natural Man*. MIT Press.

Brown, C., Cowperthwaite, A., Hijazi, A., & Somayaji, A. (2009). *Analysis of the 1999 DARPA/Lincoln Laboratory IDS evaluation data with NetADHICT; Analysis of the 1999 DARPA/Lincoln Laboratory IDS evaluation data with NetADHICT*. https://doi.org/10.1109/CISDA.2009.5356522

Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning ☆.

Information Processing and Management, *59*, 102798. https://doi.org/10.1016/j.ipm.2021.102798

Chiche, A., & Meshesha, M. (2021). *Towards a Scalable and Adaptive Learning Approach for Network Intrusion Detection*. https://doi.org/10.1155/2021/8845540

Chollet, F. (2018). *Deep learning with Python*. Manning Publications Company.

Cisco Mobile Visual Networking Index (VNI). (2017). Retrieved June 21, 2024, from https://newsroom.cisco.com/c/r/newsroom/en/us/a/y2017/m02/cisco-mobile-visual-networking-index-vni-forecast-projects-7-fold-increase-in-global-mobile-data-traffic-from-2016-2021.html

Cronquist, A. (1981). *An integrated system of classification of flowering plants*. Columbia university press.

Dadkhah, S., Mahdikhani, H., Danso, P. K., Zohourian, A., Truong, K. A., & Ghorbani, A. A. (n.d.). *Towards the Development of a Realistic Multidimensional IoT Profiling Dataset*. https://doi.org/10.1109/PST55820.2022.9851966

Daş, R., Karabade, A., & Tuna, G. (2015). *Common network attack types and defense mechanisms; Common network attack types and defense mechanisms*. https://doi.org/10.1109/SIU.2015.7130435

Devan, P., & Khare, N. (2020). An efficient XGBoost-DNN-based classification model for network intrusion detection system. *Neural Computing and Applications*, *32*. https://doi.org/10.1007/s00521-020-04708-x

Dwivedi, S., Vardhan, M., Tripathi, S., Alok, ·, & Shukla, K. (2019). Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection. *Evolutionary Intelligence*, *13*, 103–117. https://doi.org/10.1007/s12065-019-00293-8

Elmasry, W., Akbulut, A., & Zaim, A. H. (2020). Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. *Computer Networks*, *168*, 107042. https://doi.org/10.1016/j.comnet.2019.107042

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, *7*, 82512–82521. https://doi.org/10.1109/ACCESS.2019.2923640

Glover, B. (2007). *Understanding flowers and flowering: an integrated approach*.

Gogoi Prasanta and Bhuyan, M. H. and B. D. K. and K. J. K. (2012). Packet and Flow Based Network Intrusion Dataset. In D. and R. O. F. and S. R. and Y. Y. and Z. A. Parashar Manish and Kaushik (Ed.), *Contemporary Computing* (pp. 322–334). Berlin, Heidelberg: Springer Berlin Heidelberg.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Gu, J., & Lu, S. (2021). An effective intrusion detection approach using SVM with naïve Bayes feature embedding. *Computers & Security*, *103*, 102158. https://doi.org/10.1016/j.cose.2020.102158

Gu, J., Wang, L., Wang, H., & Wang, S. (2019). A novel approach to intrusion detection using SVM ensemble with feature augmentation. *Computers & Security*, *86*, 53–62. https://doi.org/10.1016/j.cose.2019.05.022

Gupta, B., Agrawal, D. P., & Yamaguchi, S. (Eds.). (2016). *Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security*. https://doi.org/10.4018/978-1-5225-0105-3

Gupta, N., Jindal, V., & Bedi, P. (2021). LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system. *Computer Networks*, *192*, 108076. https://doi.org/10.1016/j.comnet.2021.108076

Hajisalem, V., & Babaie, S. (2018). A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. *Computer Networks*, *136*, 37–50. https://doi.org/10.1016/j.comnet.2018.02.028

Halim, Z., Nadeem Yousaf, M., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., … Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*, *110*, 102448. https://doi.org/10.1016/j.cose.2021.102448

Hebb, D. (1949). The organization of behavior john wiley & sons. *New York*.

Hodo, E., Bellekens, X., Hamilton, A., Dubouilh, P. L., Iorkyase, E., Tachtatzis, C., & Atkinson, R. (2016). Threat analysis of IoT networks using artificial neural network intrusion detection system. *2016 International Symposium on Networks, Computers and Communications, ISNCC 2016*. https://doi.org/10.1109/ISNCC.2016.7746067

Hugh, J. M. (2000). *Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory*.

Hutchings, M. J., & Bell, A. D. (1991). Plant Form: An Illustrated Guide to Flowering Plant Morphology. *Journal of Ecology*, *79*(2), 557.

IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved May 28, 2024, from https://www.unb.ca/cic/datasets/ids-2018.html

Inayat, Z., Gani, A., Anuar, N. B., Khan, M. K., & Anwar, S. (2016). Intrusion response systems: Foundations, design, and challenges. *Journal of Network and Computer Applications*, *62*, 53–74. https://doi.org/10.1016/J.JNCA.2015.12.006

Jia, H., Liu, J., Zhang, M., He, X., & Sun, W. (2021). Network intrusion detection based on IE-DBN model. *Computer Communications*, *178*, 131–140. https://doi.org/10.1016/J.COMCOM.2021.07.016

Kalra, S., & Arora, S. (2016). Firefly algorithm hybridized with flower pollination algorithm for multimodal functions. *Proceedings of the International Congress on Information and Communication Technology: ICICT 2015, Volume 1*, 207–219.

Kanimozhi, V., & Jacob, T. P. (2019). Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. *ICT Express*, *5*(3), 211–214. https://doi.org/10.1016/J.ICTE.2019.03.003

Kanna, R., & Santhi, P. (2021). *Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial-Temporal Features*. *226*, 107132. https://doi.org/10.1016/j.knosys.2021.107132

KDD Cup 1999 Data. (n.d.). Retrieved June 21, 2024, from https://kdd.ics.uci.edu/databases/kddcup99/kddcup99

Kelleher, J. D., Mac Namee, B., & D'arcy, A. (2020). *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press.

Kennedy, J. (2010). Particle Swarm Optimization. In G. I. Sammut Claude and Webb (Ed.), *Encyclopedia of Machine Learning* (pp. 760–766). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-30164-8_630

Kim, J., Kim, J., Kim, H., Shim, M., & Choi, E. (2020). *CNN-Based Network Intrusion Detection against Denial-of-Service Attacks*. https://doi.org/10.3390/electronics9060916

Krishnaveni, S., Vigneshwar, P., Kishore, S., Jothi, B., & Sivamohan, S. (2020). Anomaly-Based Intrusion Detection System Using Support Vector Machine. *Advances in Intelligent Systems and Computing*, *1056*, 723–731. https://doi.org/10.1007/978-981-15-0199-9_62/COVER

Kumar, D., & Ramakrishnan, A. G. (2016). Binary classification posed as a quadratically constrained quadratic programming and solved using particle swarm optimization. *Sadhana - Academy Proceedings in Engineering Sciences*, *41*(3), 289–298. https://doi.org/10.1007/S12046-016-0466-Y/TABLES/7

Lee, H. (2016). *Clustering and Classification Methods for Prediction of the risk for Developing Disease*. State University of New York at Stony Brook.

Li, P., Chen, Z., Yang, L. T., Gao, J., Zhang, Q., & Deen, M. J. (2019). An Incremental Deep Convolutional Computation Model for Feature Learning on Industrial Big Data. *IEEE Transactions on Industrial Informatics*, *15*(3), 1341–1349. https://doi.org/10.1109/TII.2018.2871084

Li, Xin, Yi, P., Wei, W., Jiang, Y., & Tian, L. (2021). *LNNLS-KH: A Feature Selection Method for Network Intrusion Detection*. https://doi.org/10.1155/2021/8830431

Li, Xukui, Chen, W., Zhang, Q., & Wu, L. (2020). Building Auto-Encoder Intrusion Detection System based on random forest feature selection. *Computers & Security*, *95*, 101851. https://doi.org/10.1016/j.cose.2020.101851

Liu, G., Zhao, H., Fan, F., Liu, G., Xu, Q., & Nazir, S. (2022). An Enhanced Intrusion Detection Model Based on Improved kNN in WSNs. *Sensors 2022, Vol. 22, Page 1407*, *22*(4), 1407. https://doi.org/10.3390/S22041407

Liu, J., Gao, Y., & Hu, F. (2021). A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers & Security*, *106*, 102289. https://doi.org/10.1016/j.cose.2021.102289

Lv, L., Wang, W., Zhang, Z., & Liu, X. (2020). *A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine ☆. 195*, 105648. https://doi.org/10.1016/j.knosys

Marinova-Boncheva, V. (2007). *A Short Survey of Intrusion Detection Systems*.*

Mazini, M., Shirazi, B., & Mahdavi, I. (2018). *Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms*. https://doi.org/10.1016/j.jksuci.2018.03.011

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*, 115–133.

Mebawondu, J. O., Alowolodu, O. D., Mebawondu, J. O., & Adetunmbi, A. O. (2020). Network intrusion detection system using supervised learning paradigm. *Scientific African*, *9*, 497. https://doi.org/10.1016/j.sciaf.2020.e00497

Meftah, S., Rachidi, T., & Assem, N. (2019). Network Based Intrusion Detection Using the UNSW-NB15 Dataset. *International Journal of Computing and Digital Systems*, *8*(5), 2210–142. https://doi.org/10.12785/ijcds/080505

Migliavacca, M., Eyers, D. M., Shand, B., Bacon, J., & Pietzuch, P. (2010). *DEFCON: High-Performance Event Processing with Information Security*.

Miguel-Hurtado, O., Guest, R., Stevenage, S. V, Neil, G. J., & Black, S. (2016). Comparing machine learning classifiers and linear/logistic regression to explore the relationship between hand dimensions and demographic characteristics. *PloS One*, *11*(11), e0165521.

Mirjalili, S., Wang, G. G., & Coelho, L. dos S. (2014). Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. *Neural Computing and Applications*, *25*(6), 1423–1435. https://doi.org/10.1007/S00521-014-1629-6/FIGURES/13

Mitchell, T. M. (1997). *Machine learning*. McGraw-hill.

Murtugudde, G. (2021). An efficient algorithm for anomaly intrusion detection in a network. *Global Transitions Proceedings*, *2*, 255–260. https://doi.org/10.1016/j.gltp.2021.08.066

Nazir, A., & Ahmed Khan, R. (2021). *A novel combinatorial optimization based feature selection method for network intrusion detection*. https://doi.org/10.1016/j.cose.2020.102164

Nechaev, B., Allman, M., Paxson, V., & Gurtov, A. (2010). *A Preliminary Analysis of TCP Performance in an Enterprise Network *.*

Ng, A., & Jordan, M. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in Neural Information Processing Systems, 14*.

Nguyen, M. T., & Kim, K. (2020). Genetic convolutional neural network for intrusion detection systems. *Future Generation Computer Systems*, *113*, 418–427. https://doi.org/10.1016/j.future.2020.07.042

NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. (n.d.). Retrieved April 16, 2024, from https://www.unb.ca/cic/datasets/nsl.html

Oche Onah, J., Abdulhamid, M., Abdullahi, M., Hayatu Hassan, I., & Al-Ghusham, A. (2021). Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment. *Machine Learning with Applications*, *6*, 100156. https://doi.org/10.1016/j.mlwa.2021.100156

Park, H.-A. (2013). An introduction to logistic regression: from basic concepts to interpretation with particular attention to nursing domain. *Journal of Korean Academy of Nursing*, *43*(2), 154–164.

Pavlyukevich, I. (2007). Lévy flights, non-local search and simulated annealing. *Journal of Computational Physics*, *226*(2), 1830–1844.

Peppes, N., Daskalakis, E., Alexakis, T., Adamopoulou, E., & Demestichas, K. (2021). Performance of Machine Learning-Based Multi-Model Voting Ensemble Methods for Network Threat Detection in Agriculture 4.0. *Sensors 2021, Vol. 21, Page 7475*, *21*(22), 7475. https://doi.org/10.3390/S21227475

Prusty, S., Levine, B. N., & Liberatore, M. (2011). Forensic investigation of the OneSwarm anonymous filesharing system. *Proceedings of the ACM Conference on Computer and Communications Security*, 201–213. https://doi.org/10.1145/2046707.2046731

Qureshi, A.-U.-H., Larijani, H., Ahmad, J., & Mtetwa, N. (2018). *A Novel Random Neural Network Based Approach for Intrusion Detection Systems*.

Rahul, V. K., Vinayakumar, R., Soman, K., & Poornachandran, P. (2018). Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018*. https://doi.org/10.1109/ICCCNT.2018.8494096

Rao, K. N., Rao, K. V., & Reddy, P. (2021). A hybrid Intrusion Detection System based on Sparse autoencoder and Deep Neural Network. *Computer Communications*, *180*, 77–88. https://doi.org/10.1016/j.comcom.2021.08.026

Rokach, L., & Maimon, O. (2005). Decision Trees. *Data Mining and Knowledge Discovery Handbook*, 165–192. https://doi.org/10.1007/0-387-25465-X_9

Sabhnani, M., & Serpen, G. (2003). KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection. *M Sabhnani, G Serpen*. Retrieved from https://www.researchgate.net/profile/Gursel-Serpen/publication/221199497_KDD_Feature_Set_Complaint_Heuristic_Rules_for_R2L_Attack_Detection/links/58e7b6450f7e9b978f7f20db/KDD-Feature-Set-Complaint-Heuristic-Rules-for-R2L-Attack-Detection.pdf

Selvakumar B, & Muneeswaran K. (2018). Firefly algorithm based feature selection for network intrusion detection. *Computers & Security*, *81*, 148–155. https://doi.org/10.1016/j.cose.2018.11.005

Shafer, G. (1985). Conditional Probability. *International Statistical Review / Revue Internationale de Statistique*, *53*(3), 261. https://doi.org/10.2307/1402890

Shahraki, A., Abbasi, M., & Haugen, Ø. (2020). Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost. *Engineering Applications of Artificial Intelligence*, *94*, 103770. https://doi.org/10.1016/j.engappai.2020.103770

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). *Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*. https://doi.org/10.5220/0006639801080116

Sharma, N. V, & Singh Yadav, N. (2021). *An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers*. https://doi.org/10.1016/j.micpro.2021.104293

Sharmila, B. S., & Nagapadma, R. (2019). Intrusion detection system using naive bayes algorithm. *2019 5th IEEE International WIE Conference on Electrical and Computer Engineering, WIECON-ECE 2019 - Proceedings*. https://doi.org/10.1109/WIECON-ECE48653.2019.9019921

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, *31*(3), 357–374. https://doi.org/10.1016/J.COSE.2011.12.012

Singh Bhati, B., Rai, C. S., Balamurugan, B., & Al-Turjman, F. (2020). An intrusion detection scheme based on the ensemble of discriminant classifiers R. *Computers and Electrical Engineering*, *86*, 106742. https://doi.org/10.1016/j.compeleceng.2020.106742

Snapp, S. R., Brentano, J., Dias, G. V, Goan, T. L., Heberlein, T., Ho, C., … others. (1991). Dids (distributed intrusion detection system)–motivation. *Architecture, and an Early Prototype, InProceedings of the 14th National Computer Security Conference, Washington, DC*, 167–176.

Sona, A. S., & Sasirekha, N. (2021). *Withdrawal Notice WITHDRAWN: Kulczynski indexed dragonfly feature optimization based Polytomous Adaptive Base classifier for anomaly intrusion detection*. Retrieved from https://www.else-

Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., & Nakao, K. (2011). Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. *Proceedings of the 1st Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS 2011*, 29–36. https://doi.org/10.1145/1978672.1978676

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2), 130. https://doi.org/10.11919/J.ISSN.1002-0829.215044

Sperotto, A., Sadre, R., van Vliet, F., & Pras, A. (2009). A Labeled Data Set for Flow-Based Intrusion Detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5843 LNCS*, 39–50. https://doi.org/10.1007/978-3-642-04968-2_4

Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2000). *Cost-based Modeling for Fraud and Intrusion Detection: Results from the JAM Project*. Retrieved from http://www.cs.columbia.edu/

Storn, R., & Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328/METRICS

Subasi, A. (2020). *Practical machine learning for data analysis using python*. https://doi.org/10.1016/B978-0-12-821379-7.00008-4

Swarna Priya, R. M., Maddikunta, P. K. R., Parimala, M., Koppu, S., Gadekallu, T. R., Chowdhary, C. L., & Alazab, M. (2020). An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. *Computer Communications*, *160*, 139–149. https://doi.org/10.1016/J.COMCOM.2020.05.048

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A detailed analysis of the KDD CUP 99 data set; A detailed analysis of the KDD CUP 99 data set*. https://doi.org/10.1109/CISDA.2009.5356528

The UNSW-NB15 Dataset | UNSW Research. (n.d.). Retrieved May 28, 2024, from https://research.unsw.edu.au/projects/unsw-nb15-dataset

Thilagam, T., & Aruna, R. (2021). *Intrusion detection for network based cloud computing by custom RC-NN and optimization-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)*. https://doi.org/10.1016/j.icte.2021.04.006

Turing, A. M. (2009). Computing Machinery and Intelligence. In G. and B. G. Epstein Robert and Roberts (Ed.), *Parsing the Turing Test: Philosophical and*

Methodological Issues in the Quest for the Thinking Computer (pp. 23–65). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-1-4020-6710-5_3

Vasilomanolakis, E., Karuppayah, S., Muhlhauser, M., & Fischer, M. (2015). Taxonomy and Survey of Collaborative Intrusion Detection. *ACM Computing Surveys (CSUR), 47*(4). https://doi.org/10.1145/2716260

Vidal, J. M., Monge, M. A. S., & Monterrubio, S. M. M. (2020). Anomaly-Based Intrusion Detection: Adapting to Present and Forthcoming Communication Environments. *Https://Services.Igi-Global.Com/Resolvedoi/Resolve.Aspx?Doi=10.4018/978-1-5225-9611-0.Ch010*, 195–218. https://doi.org/10.4018/978-1-5225-9611-0.CH010

Vijayanand, R., Devaraj, D., & Kannapiran, B. (2018). Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection. *Computers & Security*, *77*, 304–314. https://doi.org/10.1016/j.cose.2018.04.010

Wang, Z., Liu, Y., He, D., & Chan, S. (2021). *Intrusion detection methods based on integrated deep learning model*. https://doi.org/10.1016/j.cose.2021.102177

Wu, D., Jennings, C., Terpenny, J., Gao, R. X., & Kumara, S. (2017). A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *Journal of Manufacturing Science and Engineering, Transactions of the ASME*, *139*(7). https://doi.org/10.1115/1.4036350/454654

Wu, Z., Wang, J., Hu, L., Zhang, Z., & Wu, H. (2020). A network intrusion detection method based on semantic Re-encoding and deep learning. *Journal of Network and Computer Applications*, *164*, 102688. https://doi.org/10.1016/j.jnca.2020.102688

Wutyi, K. S., & Thwin, M. M. S. (2016). Heuristic rules for attack detection charged by NSL KDD dataset. *Advances in Intelligent Systems and Computing*, *387*, 137–153. https://doi.org/10.1007/978-3-319-23204-1_15/COVER

Xanthopoulos Petros and Pardalos, P. M. and T. T. B. (2013). Linear Discriminant Analysis. In *Robust Data Mining* (pp. 27–33). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4419-9878-1_4

Xiao, M., & Xiao, D. (2007). *Alert Verification Based on Attack Classification in Collaborative Intrusion Detection*. https://doi.org/10.1109/SNPD.2007.216

Xiao, Y., Xing, C., Zhang, T., & Zhao, Z. (2019). An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks. *IEEE Access*, *7*, 42210–42219. https://doi.org/10.1109/ACCESS.2019.2904620

Xie, M., & Hu, J. (2013). Evaluating host-based anomaly detection systems: A preliminary analysis of ADFA-LD. *Proceedings of the 2013 6th International Congress on Image and Signal Processing, CISP 2013*, *3*, 1711–1716. https://doi.org/10.1109/CISP.2013.6743952

Yang, X.-S. (2012). Flower Pollination Algorithm for Global Optimization. In N. Durand-Lose Jérôme and Jonoska (Ed.), *Unconventional Computation and Natural Computation* (pp. 240–249). Berlin, Heidelberg: Springer Berlin Heidelberg.

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2019). *Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network*. https://doi.org/10.3390/s19112528

Zargar, G. R., & Kabiri, P. (2009). *Identification of Effective Network Features to Detect Smurf Attacks*. Retrieved from http://www.Tcpdump.org

Zhang, J., Ling, Y., Fu, X., Yang, X., Xiong, G., & Zhang, R. (2020). Model of the intrusion detection system based on the integration of spatial-temporal features. *Computers & Security*, *89*, 101681. https://doi.org/10.1016/j.cose.2019.101681

Zhang, Y., Wang, S., & Ji, G. (2015). A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Mathematical Problems in Engineering*, *2015*. https://doi.org/10.1155/2015/931256

Zhao, H., Li, M., & Zhao, H. (2019). *Artificial intelligence based ensemble approach for intrusion detection systems q*. https://doi.org/10.1016/j.jvcir.2019.102736

Zhou, Ying, Mazzuchi, T. A., & Sarkani, S. (2020). M-AdaBoost-A based ensemble system for network intrusion detection. *Expert Systems With Applications*, *162*, 113864. https://doi.org/10.1016/j.eswa.2020.113864

Zhou, Yuyang, Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, *174*, 107247. https://doi.org/10.1016/j.comnet.2020.107247