T.C.

KONYA TEKNİK ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DESTEK VEKTÖR MAKİNESİ İLE SİNÜS COSİNÜS ALGORİTMASI KULLANILARAK HİBRİT SALDIRI TESPİT SİSTEMİNİN TASARIMI

Salaad Mohamed SALAAD

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ocak-2021
KONYA

TEZ KABUL VE ONAYI

Salaad Mohamed SALAAD tarafından hazırlanan “Destek Vektör Makinesi (DVM) ile Sinüs Cosinüs Algoritması Kullanılarak Hibrit Saldırı Tespit Sisteminin Tasarımı ” adlı tez çalışması 29/01/2021 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Prof. Dr. Erkan ÜLKER	……..
Danışman Prof. Dr. Erkan ÜLKER	……..
Üye Dr.Öğr.Üyesi Alper KILIÇ	……..
Üye Dr.Öğr.Üyesi Özkan İNİK	……..

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Salaad Mohamed SALAAD
Tarih: 29.01.2021

ÖZET

YÜKSEK LİSANS TEZİ

DESTEK VEKTÖR MAKİNESİ İLE SİNÜS COSİNÜS ALGORİTMASI KULLANILARAK HİBRİT SALDIRI TESPİT SİSTEMİNİN TASARIMI

Salaad Mohamed SALAAD

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Erkan ÜLKER

2021, 93 Sayfa

Jüri
Prof. Dr. Erkan ÜLKER
Dr.Öğr.Üyesi Alper KILIÇ
Dr.Öğ.Üyesi Özkan İNİK

Teknolojinin gelişmesi ve internetin yaygınlaşmasıyla birlikte kritik verilerin depolanması, çevrimiçi eğitim ve çevrimiçi alışveriş gibi amaçlarla bilgisayar sistemlerinin kullanımı artmıştır. Ancak kullanıcılar sıklıkla işletim sistemlerinin kararlılığını bozan tehditlerle karşı karşıya gelmektedir. Başka bir deyişle, saldırganlar sistemlere izinsiz erişerek özel bilgilere ulaşmayı hedef almaktadırlar. Bu güvenlik açığının üstesinden gelmek için çoğu bilim insanı, saldırı tespit sistemi olarak hibritleştirilmiş yöntemlerin kullanımına dikkat çekmişlerdir. Bunun sonucunda, önerilen yöntemlerin performansını artırmak ve aynı zamanda minimum öznitelikleri seçmek için meta-sezgisel algoritmalar ve makine öğrenimi yöntemleri gibi çeşitli teknikler sunulmaktadır. Bu tezde, yüksek performans elde etmek ve aynı zamanda doğruluğu artırmak için makine öğrenimi algoritmalarından seçilen iki başarılı algoritmanın (Binary Sinüs kosinüs Algoritması (BSCA) ve Destek Vektör Makinesi (DVM)) entegre edilmesi hedeflenmiştir. Özellik seçimindeki başarısından dolayı BSCA nın özellik seçimi için kullanılması ve sınıflandırmadaki başarısından dolayı da SVM algoritmasının sınıflandırıcı olarak kullanılması öngörülmüştür. Bu tez çalışmasının amacı, makine öğrenmesi algoritmalarından hibrit olarak oluşturulan yeni algoritmanın saldırganları tespit etme potansiyelini vurgulamak ve en uygun hibrit saldırı tespit sistemini elde etmektir. Sunulan modelin performansını değerlendirmek için 2 farklı veri seti kullanılmıştır. Ayrıca, diğer makine öğrenme tekniklerinin kullanımları ve başarı oranları da dikkate alınarak önerilen yeni hibrit sistemin; Radial Basis Function (RBF) ve Polinom çekirdekli Destek Vektör Makinesi (RBF-DVMve Polinom DVM), Rastgele Orman (RO), K-en yakın komşu (k-NN), Naive Bayes sınıflandırıcısı (NBC), Binary Parçacık Sürü Optimizasyonu entegreli DVM (BPSO-DVM) ve literatürden seçilmiş güncel mevcut bazı çalışmalarla performans karşılaştırmaları (IWD-SVM, GA-SVM, MBGW-SVM, LOA-CNN gibi) yapılmıştır. NSL-KDD ve UNSW-NB15 veri setlerini kullanarak, önerilen yöntem sırasıyla % 99,30 ve % 99,70 doğruluğa elde edildi.

Anahtar Kelimeler: Destek Vektör Makinesi, Hibrit Saldırı Tespit Sistemi, İkili Sinüs Kosinüs Algoritması, NSL-KDD, UNSW-NB15, Öznitelik Seçimi, Sınıflandırma

ABSTRACT

MASTER THESIS

THE DESIGN OF HYBRID INTRUSION DETECTION SYSTEM BY USING SINE COSINE ALGORITHM WITH SUPPORT VECTOR MACHINE

Salaad Mohamed SALAAD

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Advisor: Prof. Dr. Erkan ÜLKER

2021, 93 Pages

Jury
Prof. Dr. Erkan ÜLKER
Dr.Öğr.Üyesi Alper KILIÇ
Dr.Öğ.Üyesi Özkan İNİK

Nowadays the use of computer systems became terrifically significant due to advance technology, simplicity and availability of the internet. Moreover, the computer systems play a crucial role for our daily life in various objective such as; storing critical data, online education and online shopping. Nevertheless, users of the computer systems frequently encounter threats which undermines the maintenance and stability of their operating systems. Particularly, politician systems, banks and etc. the intruders attempt to obtain the individual information by unauthorized access to the private systems. To accomplish this vulnerability, majority of scientists have attracted their attention the use of hybridized methods as intrusion detection systems. Therefore, diverse techniques like meta-heuristic algorithms and machine learning methods are proposed in order to boost the performance of the presented methods as well to select minimum attributes. In this thesis, it is goaled to combine two successful algorithms selected from meta-heuristic and machine learning algorithms (Binary Sine Cosine Algorithm (BSCA) and Support Vector Machine (SVM)) to attain superior performance and simultaneously enhance accuracy. It foresaw to use BSCA for feature selection for its success in feature selection and to use SVM algorithm as a classifier due to its success in classification. The main objective this thesis is to highlight the implicit of a new algorithm, which is integrated as a hybrid form machine learning algorithm, in order to detect attackers and to acquire the most acknowledge hybrid intrusion detection systems. Two different data sets were used to evaluate the performance of the presented model.The result of the new proposed hybrid system will be compared to some success rates of other machine learning techniques such as; RBF and Polynomial Support Vector Machine (RBF-SVM and Polynomial SVM), Random Forest (RF), K-nearest neighbor (k-NN), Naive Bayes classifier (NBC), SVM with Binary Particle Swarm Optimization (BPSO-SVM). And some current existing studies (such as IWD-SVM, GA-SVM, MBGW-SVM and LOA-CNN) which is selected from the literature.Using NSL-KDD and UNSW-NB15 datasets, the proposed method achieved 99.30% and 99.70% accuracy respectively.

Keywords: Hybrid Intrusion Detection System, Binary Sine Cosine Algorithm, Support Vector Machine, NSL-KDD, UNSW-NB15, Feature Selection, Classification.

ÖNSÖZ

Bu araştırmanın temel konsepti, önerilen yöntemlerin performansını artırmak ve minimum özellikleri seçmek için hibrit saldırı tespit sisteminin tasarımıdır. Bu tezde, aynı zamanda yüksek performans elde etmek ve doğruluğu artırmak için makine öğrenme algoritmalarından seçilen algoritmaların (Sinüs kosinüs Algoritması (SCA) ve Destek Vektör Makinesi (SVM)) entegre edilmesi amaçlanmıştır.

Öncelikle Allah'a sonsuz teşekkür ederim. İkincisi, beni destekleyen birkaç unutulmaz insana teşekkür etmek istiyorum. Araştırma sürecimdeki yardımı, minnettarlığı, motivasyonu, değerli yorumu, coşkusu ve üstün rehberliği için danışmanım Prof. Dr. Erkan Ülker'e son derece minnettarım ve kendisine borçluyum.

En büyük ilham kaynağım kardeşlerime Daauud Mohamed SALAAD, Zakaria Mohamed SALAAD ve Abdullah Mohamed SALAAD'a doğrudan ve dolaylı olarak mutlak yardımları için içtenlikle teşekkür ederim. Liseden bu yana bana cesaret vermesi ve dostluğu nedeniyle yakın arkadaşım Ali Diriye JIMALE'ye de teşekkürü borç bilirim.

Tabii ki, arkadaşım Abdullah Hüseyin ABDULLAHİ'nin dürüstçe tavsiyesi, nazik desteği ve çalışmamla ilgili açık rehberliğinden burada bahsetmek olağanüstü ve memnuniyet verici.

Nihayetinde, ailemdeki her bir bireyin nazik, sevgi dolu ve birçok yardımları için sonsuz minnettarım. Geleceğim için çalışmaya teşvik ettikleri ve eğitimimde elde ettiğim her başarılı adımda da benimle gurur duyduklarından dolayı ebeveynlerim Mohamed Salaad IBRAIM ve Nuunaay Ahemd HAMUD'a özel ve en içten teşekkürlerimi ifade etmek istiyorum.

Salaad Mohamed SALAAD
KONYA-2021

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Simgeler

@ : “at / güzel a” simgesi
Θ : theta
∑ : Sigma işareti (Operandların Kümülatif Toplamını ifade eder)

Kısaltmalar

ABC : Yapay Arı Kolonisi (Artificial Bee Colony).
ACCS : Avustralya Siber Güvenlik Merkezi (Australian Centre for Cyber Security).
ANN : Yapay Sinir Ağı (Artificial Neural Network).
ASCA-PSO : Parçacık sürüsü ile adaptif sinüs kosinüs algoritması (Adaptive Sine cosine algorithm with particle swarm optimization).
BMM-ADS : Beta Karışım Tekniğiile Anomali tespit sistemi (beta mixture technique with - Anomaly detection system)
BPSO : İkili Parçacık Sürü Optimizasyonu (Adaptive Sine cosine algorithm with particle swarm optimization).
CIA : Gizlilik, Bütünlük, Kullanılabilirlik (Confidentiality, Integrity, Availability).
CNN : Evrişim Sinir Ağı (Convolution Neural Network).
DE : Diferansiyel evrim (Differential evolution).
DOS : Hizmet Reddi (Denial of Service).
DT : Karar Ağacı (Decision Tree).
FCM : Bulanık C-Kümelenme Demektir (Fuzzy C-Means Clustering).
FN : Yalancı Negatif (False Negative).
FPR : Yalancı Pozitiflik Oranı (False Positive Rate).
FS : Öznitelik Seçimi (Feature Selection).
GA : Genetik Algoritma (Genetic Algorithm).
GA-SVM : Genetik Algoritma ile Destek Vektör Makinesi (Genetic Algorithm with Support Vector Machine).
GOSA :Çekirge Optimizasyon Algoritması ve Benzetilmiş Tavlama (Grasshopper Optimization algorithm and simulated annealing).
HIDS : Host Saldırı tespit sistemi (Host Intrusion Detection System).
IDS : Saldırı Tespit Sistemi (Intrusion Detection System).
IWD : Akıllı Su Damlaları (Intelligent Water Drops).
K-NN : K en yakın komşuluk (K nearest neighborhood).
LOA : Aslan Optimizasyonu Algoritması (Lion Optimization Algorithm).
LSP+CFNN : Konvolüsyonlu Sinir Ağları ile Locust Swarm Optimizasyonu (Locust Swarm Optimization with Convolution Neural Networks).
LSVM : Doğrusal Destek Vektör Makinesi (Linear Support Vector Machine).
MBGWO : Modifiye İkili Gri Kurt Optimizasyonu (Modified Binary Grey wolf optimization).
ML : Makine öğrenmesi (Machine Learning).

NBC : Naif Bayes Sınıflandırma (Naïve Bayesian Classification).
NIDS : Ağ Saldırı tespit sistemi (Network Intrusion Detection System).
RBF : Radyal Temel Fonksiyon (Radial Basic Function).
RF : Rastgele Orman (Random forest).
R2L : Yerel uzaktan (Remote to Local).
SCA : Sinüs Kosinüs Algoritması (Sine Cosine Algorithm).
BSCA-SVM : Sinüs Kosinüs Algoritması ile Destek Vektör Makinesi (Sine Cosine Algorithm with Support Vector Machine).
SVM : Destek Vektör Makinesi (Support Vector Machine).
TP : Gerçek Pozitif (Tue Positive).
U2R : Kök Kullanıcı (User to Root).

1. GİRİŞ

1.1. Tezinn Amacı

Siber güvenlik, bilgi güvenliğinde en önemli ve endişe verici konulardan biri haline gelmiştir. Son yıllarda araştırmacılar güvenlik sisteminin hem ana bilgisayarlarda hem de ağdaki güvenlik açıklarıyla ilgilenmesi gerektiğini belirtmişlerdir. Bu iki kanal, saldırganları güçlendirmekte ve güvenlik politikasına aykırı davranmak için sistemlere yetkisiz bir şekilde erişmek için önemli bir şans vermektedir. Diğer yandan hızlı şekilde yayılan internet ve hızlı gelişen teknoloji, saldırganların yasadışı faaliyetleri için başka fırsatlar doğurmuştur.

Çoğu uzman, saldırganların hareketlerini yanıtlayıp engelleyerek, aykırılık eylemini önleme yeteneğine sahip sağlam bir model (araç) oluşturmayı önermiştir. Girişimlerden biri, etkili bir saldırı tespit sistemi bulmak ve yaratmaktır. Bunun neticesinde, saldırı tespit sistemi modelleri oluşturmak için çok sayıda araştırmacı, kendini farklı alanlarda ispatlamış olan makine öğrenmesi yöntemlerini kullanmışlardır. Metasezgisel (stokastik) algoritmalar günümüzün en etkili ve ilgi gören alanlarından biridir. Metasezgisellerin yardımıyla gerçeklenecek olan özelliklerin azaltılması vasıtasıyla modele tespit için verimli bir performans garantisi sağlanacaktır. Aynı zamanda sınıflandırma sonuçları için üstün performans elde etmek amacıyla sınıflandırıcı algoritmasını iyileştirmeyi de mümkün kılacaktır. Bu nedenle tez çalışmasının öncelikli amacı, Sinüs Kosinüs Algoritması (SKA) ve Destek vektör makinasına (DVM) dayalı yeni bir hibrit saldırı tespit sistemi modeli oluşturmak ve sunmaktır.

Çalışmada özet olarak özellik seçimi için Sinüs Kosinüs algoritmasına (SKA) dayalı yeni bir hibrid saldırı tespit sistemi geliştirmek ve sınıflandırma amacıyla da Destek Vektör Makinası (DVM) kullanmak amaçlanmıştır. SCA ile SVM kombinasyonuyla üretilecek hibrit algoritmanın, performans değerlendirmesini daha doğru yapabilmek için diğer metasezgisel algoritmalarla karşılaştırmalar da yapılmıştır.

Temel amaç, saldırganları etkin bir şekilde saptayabilen ve performansıyla nispeten başarılı bir çözüme ulaşabilen yeni bir hibrit saldırı tespit modeli elde etmektir. Ek olarak, sunulan yeni sistemin, birçok saldırı tespit araştırmacısı tarafından kabul görmüş UNSW-NB15 ve NSL-KDD veri setleri kullanılarak performans testleri gerçekleştirilmiştir.

1.2 Tezin Önemi

Saldırı tespit sistemlerine doğrudan uygulanamayan birçok yüksek boyutlu veri kümesi mevcuttur. Modeller doğrudan çok boyutlu problemlere uygulanmaya çalışılırsa, sınıflandırma performansının doğruluğu düşebilir. Bu nedenle veri setinin çok boyutluluğunu azaltmak ve en önemli özellikleri seçmek gerekir. Bu zorluğa çözüm bulmak için özellik seçimi yöntemi uygulanmalıdır.

Bazı mevcut özellik seçme algoritmalarının, sınıflandırma doğruluğunun sonucunu etkileyen pahalı hesaplama maliyeti gibi zorlukları olduğu için, literatürde yazarların çoğu özellik seçimi için metasezgisel algoritmalar kullanmıştır. Bu sorunu gidermek için birçok araştırmacı, SCA algoritması (Sine Cosine Algorithm (SCA)) gibi evrimsel ve metasezgisel algoritmalar kullanarak minimum hesaplama maliyeti sağlamışlardır.

Sinüs kosinüs algoritması popülasyon tabanlı bir metasezgisel algoritmadır ve sinüs kosinüs formülünden türetilmiştir. Sinüs kosinüs algoritmasının tezde kullanılmasının amacı, orijinal veri seti özelliklerini kaybetmeden, azaltılmış ve en önemli özelliklerin elde edilebildiği bir özellik seçme mekanizmasını mümkün kılmasıdır. Sinüs kosinüs algoritmasının seçtiği özellikler, en uygun sonuçlar elde etmek için sınıflandırıcının performansını destekleyecektir.

Sınıflandırma için Destek Vektör Makinasının (DVM) kullanımı öngörülmüştür. Destek Vektör Makinası denetlenen bir makine öğrenme algoritmasıdır. DVM, sınıflandırma ve regresyon problemleri için sıklıkla kullanılmaktadır. Ayrıca, başarılı sonuçlar ve en yüksek doğruluk için en iyi sınıflandırma yöntemlerinden birisidir. Tezde, verimli ve üstün sonuçlar alması nedeniyle sınıflandırma için Destek Vektör Makinası kullanılmıştır. Ayrıca SCA tarafından seçilmiş olan özellikler, SVM algoritmasında sınıflandırma için uygulanmıştır. Seçilen özelliklerle, Destek Vektör Makinasının daha az zaman ve optimum sonuçlarla gerçekleştirilen bir sınıflandırma doğruluğuna ulaşması beklenmiştir.

Tez çalışmasında diğer makine öğrenmesi tekniklerinin de kullanımları ve başarı oranları dikkate alınmıştır. Yapılan çalışmalarından sonra, özellik seçimindeki başarısından dolayı İkili SCA nın özellik seçimi için iyi bir tercih olduğu ve sınıflandırmadaki başarısından dolayı da SVM algoritmasının kullanılmasının başlangıç olarak tercih edilmesi gerektiği öngörülmüştür. Bu tez çalışmasının amacı, makine öğrenmesi algoritmalarından seçilen bir kısmının saldırganları tespit etme potansiyelini

vurgulamak ve başarılı olan yöntemlerin kullanıldığı bir optimum hibrit saldırı tespit sistemi elde etmektir.

2. KAYNAK ARAŞTIRMASI

Birçok araştırmacı, güçlü ve iyi sınıflandırma performansına sahip bir model elde etmek için uygun bir algoritmanın seçilmesi gerektiğini öne sürmektedir. Yine bu tür sistemlere sahip olmak için birçok araştırmacı, özellikle makine öğrenmesi ve meta-sezgisel optimizasyon algoritmaları gibi farklı algoritmaları birleştiren bir karma yöntem kullanarak dikkat çekici sonuçlara ulaşabilmiştir. Literatür incelemesinde ilk olarak çalışmada kullanılması düşünülen Sinüs Kosinüs Algoritması kaynaklarına yer verilmiştir.

Hafez ve ark, Sinüs Kosinüs algoritması (SCA) ve K-en yakın komşuluk algoritmasının (KNN) bir kombinasyonu olan bir sistem sunmuşlardır. SCA yöntemi, orijinal veri kümesi özelliklerini kaybetmeden veri setinin özelliklerini en aza indirgemek amacıyla özellik seçimi için kullanılmıştır. Önerilen yöntemi değerlendirmek için UCI makine öğrenim deposu web sitesinden alınmış olan 18 veri seti incelenmiştir. Deneysel sonuçlara göre önerilen model, parçacık sürü optimizasyonu ve genetik algoritma (GA) gibi mevcut diğer stokastik arama algoritmalarına kıyasla daha iyi performans göstermiştir (Hafez ve ark, 2016).

Diferansiyel evrim (DE) ve Logistic Regresyon sınıflandırıcılı sinüs kosinüs algoritmaları da (SCA) önerilmiştir. Sınıflandırıcının daha iyi sonuç alması için özellik seçiminde SCA ile DE kullanılmıştır. Lojistik regresyon sınıflandırıcısı, modelin 8 farklı veri seti ile performansını değerlendirmek için kullanılmıştır. Makalede, Sinüs Kosinüs algoritmasının bu zorluğu gidermek için yerel optimumda sıkışıp kaldığı ve SCA'nın yerel arama sürecini atlamasına yardımcı olması için DE modelini yerel aramada kullandıkları belirtilmiştir. Lojistik regresyon sınıflandırıcısı, modelin 8 farklı veri seti ile performansını değerlendirmek için kullanılmıştır (Mohamed ve ark., 2017).

ASCA-PSO (Parçacık sürü optimizasyonuna sahip uyarlamalı sinüs kosinüs algoritması) adlı yeni bir geliştirme sürümü modeli, Aboul ve ark. (2018) tarafından önerilmiştir. Yöntemin baskınlığını kanıtlamak için, yöntemin performansı birkaç unimodal ve multimodal standart fonksiyonlar kullanılarak incelenmiştir. Yeni model, sinüs kosinüs algoritması ve diğer metasezgisel algoritmaların aksine, deneysel değerlendirmenin sonucu, bu yöntemin doğruluk ve hesaplama süresi açısından daha iyi sonuçlar elde ettiğini göstermiştir (Aboul ve ark., 2018).

Chiwen ve ark (2018), SCA kullanarak yeni bir model sunmuşlardır. Amaçları, sinüs kosinüs algoritmasının işlevselliğini, Komşu Arama ve Açgözlü Levy Mutasyonu

ile değiştirerek geliştirmektir. SCA nın temelinin, hoş olmayan çözümlere ulaşmak, düşük yakınsama hızı gibi bazı global optimizasyon zorluklarıyla uğraştığını dile getirmişlerdir. Küresel keşif ve yerel arama yeteneklerini dengelemek için hem üstel dönüşüm parametresini hem de atalet ağırlığı azaltımını kullanmışlardır. Algoritmayı basit bir şekilde yerel optimumdan kurtarmak ve arama yeteneğini geliştirmek için, birincil algoritmada optimal bireyleri değiştirmek için optimal bireylere yakın rasgele bireyler kullanmışlardır. Yine yerel arama kabiliyetini arttırmak için açgözlü Levy tekniği de kullanılmıştır Chiwen ve ark (2018)**.**

Ekiz ve ark. (2017), SCA kullanarak kısıtlı optimizasyon problemini çözmüşlerdir. Sonuçları Parçacık Sürü Optimizasyonu (PSO) ve Genetik Algoritma (GA) etkileşimi gibi iyi bilinen algoritmalar ile karşılaştırmışlardır, SCA'nın PSO ve GA hızına göre daha iyi olduğunu gözlemlemişlerdir. SCA nın optimum bir çözüme ulaşmadığı, bunun aksine, kısıtlayıcı olmayan optimizasyon problemi ile iç içe geçmiş bir sonuç elde ettiğini göstermişlerdir (Ekiz ve ark,2017).

Vijay ve Dinesh (2018), SCA kullanarak kümelenme zorluklarını çözmek için yeni bir model geliştirmişlerdir. Yeni sistemlerine sinüs kosinüs küme algoritması (SCAC) adını vermişlerdir. Sunulan yöntem, kodlama şemasına göre küme merkezini kullanır. Değerlendirme performansları için en popüler dört veri setini kullanmışlar ve yöntemlerini diğer kümelenmelerle karşılaştırmışlardır. Karşılaştırma deney sonuçları, yeni modelin diğerlerinden daha iyi olduğunu göstermiştir (Vijay ve Dinesh, 2018).

Sihag (2018), küresel sayısal optimizasyon problemleri için genel adaptif sinüs kosinüs algoritması denilen sağlam bir model önermişlerdir. Trigonometrik fonksiyonlara dayanan yeni metasezgisel algoritmanın verimli ve iyi sonuçlar elde ettiği gözlemlenmiştir (Sihag, 2018).

Hathiram ve Ravi (2017) tarafından, optimizasyon problemlerini izlemek için Diferansiyel evrim modeli ile birleştirilen SCA adlı yeni bir optimizasyon metodu önerilmiştir. Sistemlerin performansını test etmek için 23 test fonksiyonu kullanılmıştır. Deneysel sonuçlarına göre, yeni modelin performansının bazı mevcut ve iyi bilinen algoritmalara kıyasla iyi olduğuna, ek olarak, yeni sistemin keşif ve keşif süreçlerinde üstün olduğuna değinmişlerdir (Hathiram ve Ravi, 2017).

Sahlol ve Ewees (2016), Yapay Sinir ağının eğitilmesi sürecinde SCA'yı kullanmışlardır. YSA'nın ağırlıklarının optimum değere ulaşması için SCA ile ağı eğitmekte ve güncellemektedirler (Sahlol ve Ewees, 2016) .

SCA'nın zaman serisi tahmini için Destek Vektör Regresyon modeli ile bir araya getirilmesi Sai ve ark. (2017) tarafından sunulmuştur. Zaman serisi, esas olarak büyük miktarda duyusal verilere dayanan ve sürmekte olan sistem arızasını en aza indirgemeyi gerektiren bir kısım veri içermektedir. SCA, Destek Vektör Regresyonunda ceza ve çekirdek parametrelerinin seçilmesinde, sistemin performansını ilerletmek için kullanılmıştır (Sai ve ark., 2017).

SCA nın yerel arama eksikliği nedeniyle, Mohamed ve Diego (2017) tarafından muhalif temelli öğrenmeye odaklanan bir SCA türevi önerilmiştir. Arama alanında tercih edilen keşifleri yapmak için bir strateji mekanizması kullanmışlardır. Sistemin performansını test etmek için çeşitli kıyaslama fonksiyonları kullanılmıştır (Mohamed and Diego, 2017).

Saldırı tespit sisteminin performansını artırmak için SVM ile değiştirilmiş ikili gri kurt optimizasyonu (MBGWO) entegrasyonu önerilmiştir. MBGWO, sınıflandırma değerlendirmesini iyileştirmek ve yüksek doğruluk gibi daha yüksek dereceli bir sonuç elde etmek için en uygun eğitim veri özelliklerini seçmek üzere kullanılmıştır. Deneysel sonuçların doğruluğu %99,22 olarak elde edilmiştir. Bu çalışmada NSL-KDD ağ saldırı veri seti kullanılmıştır. Sunulan model, 41 boyut özelliğini 14'e düşürmüştür (Qusay ve ark., 2019).

Aslan metasezgisel optimizasyonu ve Konvolüsyon Sinir Ağlarına (CNN) dayalı yeni bir saldırı tespit sistemi önerilmiştir. Deneysel sonuçlarına göre, NSL-KDD verileri kullanılarak %96 oranında bir doğruluk sonucuna ulaşılmış ve bu modelin saldırganları tespit etmek için etkin olduğu ortaya konulmuştur. (Arivudainam ve ark., 2018).

Acharya ve ark. (2017), destek vektör makinesi (SVM) ile birleştirilen akıllı su damlacıklarına (IWD) dayanan bir hibrit kullanım saldırı tespit sistemi önermişlerdir. Bu araştırmada IWD, özellik seçimi için, SVM ise sınıflandırma performansı için kullanılmıştır. Sistem performanslarını göstermek için KDD CUP's 99 veri seti tercih edilmiştir. 41 adet özellik 9 özelliğe indirgenmiştir ve doğruluk değeri %99.0915 olarak rapor edilmiştir (Acharya ve ark., 2017).

İlyas ve ark. (2019) ileri beslemeli sinir ağı (feed-forward neural network (FNN)) ile metasezgisel yer çekimi sürüsü optimizasyonuna dayanan yeni bir yöntem tasarlaımştır. Özellik seçimi için çekirge sürüsü optimizasyonu, sınıflandırma için FNN kullanılmıştır. Bu çalışmada NSL-KDD ve UNSW-NB15 veri setlerine yer verilmiştir. Elde edilen doğruluk sonucu %95,42 olmuştur.

Filtre tabanlı bir özellik seçimi algoritması ile entegre ileri beslemeli derin sinir ağlarını (FFDNN) temel alan bir yöntem Sydney ve Yanxias (2019) tarafından önerilmiştir. Sınıflandırma performansını kontrol etmek için NSL-KDD veri seti kullanılmıştır. Yapılan çalışmanın sonucu olarak doğruluk yüksek çıkmıştır. Önerilen model, bazı popüler makine öğrenme algoritmaları olan; SVM, KNN, DT ve NBC ile karşılaştırılmıştır **(** Sydney ve Yanxıa,2019)**.**

Yapay arı kolonisi (ABC) ile bütünleşen destek vektör makinesine (SVM) dayalı bir hibrit saldırı tespit sistemi Bahareh ve ark. (2014) tarafından sunulmuştur. Ağın saldırı tespit yeteneğini geliştirmek için özellikleri azaltmak için ABC ve en iyisini seçmek için SVM kullanılmıştır. Çalışmada, ABC üstün sonuçlar elde etmek için iyi sınıflandırma performansı sunmuştur. Çalışan sistemi test etmek için KDDcup99 veri seti kullanılmış ve deneyler bu veri seti üzerinde gerçeklenmiştir. Önerilen yöntem, %99,71 doğrulukla kötü niyetli etkinlikleri önleyebilmekle birlikte, U2R ve R2L veri setlerinde kötü niyetlileri tespit etmek için iyi sonuçlar vermemiştir.

Shokoohsaljooghi ve ark. (2019), Parçacık Sürü Optimizasyonunun (PSO) sinir ağları ile entegrasyonu olan yeni bir model önermişlerdir. Bu yöntemin performansını değerlendirmek için NSL-KDD, KDD-CUP99 ve CIDD veri setleri kullanılmış ve önerilen algoritma, saldırı özelliklerini etkili bir şekilde sınıflandırıp, aynı zamanda yanlış alarmı azaltıp, algılama oranını arttırmıştır. Bu yöntemle KDD-CUP99 için %98,10 ve NSL-KDD için %99,99 doğruluk değeri elde edilmiştir (Shokoohsaljooghi ve ark., 2019).

Yapay Arı Kolonisi (ABC) ve Yapay Balık Sürsüsü (AFS) algoritmalarının yeni bir hibrit modeli Hajisalem ve Shahram tarafından önerilmiştir. Bulanık C-Means Kümeleme (FCM) ve Korelasyona Dayalı Özellik Seçimi (CFS) de eğitim verilerini bölmek ve önemsiz özellikleri çıkarmak için gerçeklenmiştir. Sunulan yeni yöntemin performansını test etmek için NSL-KDD ve UNSW-NB15 veri setleri kullanılmıştır. Sonuç olarak %99 doğruluk oranına erişilmiştir (Vajiheh Hajisalem and Shahram, 2018) **.**

3. MATERYAL VE YÖNTEM

3.1. Saldırı Tespit Sistemi Kavramı

Bilgisayar ağ kullanımının hızlı gelişmesi, ağların hızla büyümesi ve kullanıcıların birçok hassas özel veriyi internet üzerinden paylaşması nedeniyle bilgisayar saldırganlarının çoğu, ağ üzerinde yer alan bilgilerin güvenliğini ihlal etmeye çalışmaktadırlar. Bu tehditleri tespit etmek ve bu saldırıların üstesinden gelmek için güvenli bir ağa sahip olmak zorunluluğu oluşmuş, bu nedenle de saldırı tespit sistemlerinin kullanımına ihtiyaç duyulmuştur. Bilgisayar sistemi güvenliği, "Gizlilik", "Veri Bütünlüğünü Koruma" ve "Kullanılabilirlik" olmak üzere üç önemli bileşenin korunması (tespit edilmesi) süreci olarak açıklanmıştır. Bu da bilgisayarın herhangi bir zararlı faaliyeti önleme gücüne sahip olmasını sağlamaktadır (Kevr ve ark., 2016).

İzinsiz giriş, bilgisayar ağ altyapısının işlevselliğini ihlal eden yasadışı veya anormal etkinliklerin başında gelmektedir. Saldırı tespit sistemi, sisteme yetkisiz bir şekilde erişmeye çalışan herhangi bir eylemi tanımlamak için sistem etkinliklerini vurgulayan bir araç olarak tanımlanmıştır. İzinsiz giriş, veri bütünlüğünden, gizlilikten ve sistem kullanılabilirliğinden taviz vermeyi amaçlayan girişimlerin tamamı olarak tanımlanmaktadır (Kevric ve ark., 2016).

Acharya ve Singh (2017) saldırı tespit sisteminin donanım ve yazılımın bir kombinasyonu olan önemli bir siber güvenlik aracı olduğunu belirtmişlerdir. Saldırı Tespit Sistemleri, bir ağda veya ana bilgisayarda gelen trafik paketlerini analiz etmek ve normal veya anormal etkinlikler olup olmadığını tanımayı ve ayırmayı sağlamaktadır. Aslında, davetsiz misafirleri tespit etmek için sınıflandırma performansını iyileştirmeyi düşünen saldırı tespit sistemi'nin uygulanması bir sınıflandırma süreci olarak ifade edilebilir (Acharya ve Singh, 2017).

Saldırı tespit sistemi, ana bilgisayarın etkinliğini izleyen ve sürekli veri toplayan, elde ettiği verileri güvenlik sisteminde açık oluşturacak eylemlerin meydana gelip gelmediğini tespit etmek için işleyerek geriye bilgi döndüren akıllı bir sistemdir (Malik ve Khan, 2017). Saldırı tespit sisteminin birincil amacı, gizlilik, kullanılabilirlik ve veri bütünlüğü olmak üzere gerekli güvenlik esaslarını ortaya koymaktır. Gizlilik, bilgilerin yetkili kişi veya sertifikalı kişiler tarafından özel bir şekilde görüntülenebilmesini ifade etmektedir. Veri bütünlüğü, verilerin nedensiz bir şekilde değiştirilememesi veya yok edilememesi anlamına gelmektedir. Kullanılabilirlik de,

sistemin her an etkin ve yetkili kişiler tarafından herhangi bir güçlükle karşılaşılmadan ulaşılabilir olması anlamı taşımaktadır. Bu üç güvenlik esası arasında kullanılabilirliğin en önemli olması dikkat çekicidir (Shailendra and Sanjay, 2009).

Ayrıca, şifreleme yöntemi, güvenlik duvarları ve kullanıcı kimlik doğrulaması gibi işlemler geleneksel korumalar olarak nitelendirilir, ancak bu teknikler saldırı tespit sisteminin yaptığı gibi anormal eylemleri izleyemez ve engelleyemez. Bu nedenledir ki saldırı tespit sistemi, her zaman ağ katmanındaki güvenlik duvarından sonra konur. Saldırı tespit sistemi geleneksel koruma güvenlik duvarlarından daha fazla yapay zekaya ihtiyaç duymaktadır. Örneğin belirli bir paket analizi prosedürü düşünüldüğünde, bu analizde paketin bileşenleri, paket kaynağının IP adresleri ve hizmetler, bayraklar vb. gibi paketin çeşitli bölümlerini dikkate alır (Hajisalem ve Shahram, 2018).

Saldırı tespit sistemi yöntemi, sistemde meydana gelen kötü amaçlı etkinlikleri tanır veya tanımaya çalışır ve bazı istenmeyen aktivitelerin sistem güvenliğinin istikrarını ihlal etmek için çaba sarf ettiğini bildirmek için yöneticileri hemen uyarır. Saldırı tespit sisteminde yanlış pozitif ve yanlış negatif olmak üzere iki tür alarm vardır. Yanlış pozitif, normal aktivitenin bir saldırı olarak muamele görmesidir, yanlış negatif ise yanlış pozitifin tersidir. Bu iki alarmın entegrasyonu yanlış alarm oranı olarak sınırlandırılır ve saldırı tespit sisteminin performansını tahmin etmek için kullanılır (Shahri ve ark., 2015).

Aynı zamanda saldırı tespit sistemi, bilgisayar ve olay kullanıcı eylemlerindeki etkinlikleri izleyen ve analiz eden bir modeldir. Buna ek olarak, saldırı tespit sistemi sistem güvenlik açıklarını analiz eder ve ardından yapılandırmaya çalışır. Olayın normal (desen) olduğunu veya bilgisayar sistemi ve ağ ortamı içinde çeşitli kaynaklarda bilgi toplayarak yapılan bir saldırı olduğunu belirlemeye çalışır. Ayrıca güvenlik ilkesini ihlal etmeye çalışan tüm kullanıcıları da izler (Ashoor ve Gore, 2011).

Saldırı tespit sistemi, yasal denetim mekanizmalarını toplayarak ve analiz ederek kötü niyetli faaliyetleri koruyan bilgi güvenliği için gerekli bir araçtır. Meşru faaliyet ile yetkisiz olanı ayırt etmek için farklı kurallar uygular (Elhag ve ark, 2017). Saldırı tespit sistemi, bir saldırının gerçekleşmesinden önce ve sonra davetsiz misafirleri belirlemeye ve önlemeye çalışan bir modeldir. Ayrıca saldırı tespit sistemi, ağda veya bilgisayarlarda meydana gelen olayları izleyen donanım ve yazılım sistemlerinin birleşimidir (Aydın ve ark., 2009).

3.2. Saldırı Tespit Sistemine Genel Bakış

Saldırı tespit sisteminin amacı, özel verilerimizi yetkisiz kişilerin normal veya kötü niyetli faaliyetini tanımlayarak tespit etmektir. Saldırı tespit sistemi kavramı son otuz kırk yıldır literatürde yer almaktadır. James P. Anderson 1980'de "Bilgisayar güvenliği tehdidi, İzleme ve Gözetleme" başlıklı bilimsel makalesini sunduktan sonra, fikir bütün güvenlik alanında genişlemiştir (Anderson, 1980). Bundan sonra Saldırı tespit sistemi, güvenlik bilgisi alanında yerini almış ve tanınmaya başlamıştır.

Günümüzde saldırı tespit sistemi ihtiyaç duyulan bir hal almış ve güvenlik altyapısının tamamı ile ilişkilendirilmiştir. Daha sonra saldırı tespit sistemi, araştırmacıların bu konuya zaman harcaması ve gelişen teknoloji ile çeşitli aşamalardan geçmiştir. (Ashoor ve Gore, 2011).

1980'de James Anderson tarafından önerilen ve denetim yollarının önerisine açıklık getiren etkili makale, yanlış kullanım ve kullanıcı davranışlarının anlaşılmasının izlenmesinde değerli olabilecek önemli talimatlar içermektedir. "Tespit etme" kavramı ve yanlış tespit edilen kullanıcı etkinliği ortaya çıkmıştır. Bu kavram, her işletim sisteminin bir güce sahip olmasını sağlamıştır. Buna ek olarak, James Anderson'a ait bu fikir, gelecekteki temel saldırı tespit sistemlerinin tasarımına ve gelişimine yön vermiştir. Sonunda çalışması, ilk ana bilgisayar tabanlı saldırı tespit sistemi olarak ve kendisi de genel anlamda saldırı tespit sistemlerinin kurucusu olarak kabul edilmiştir.

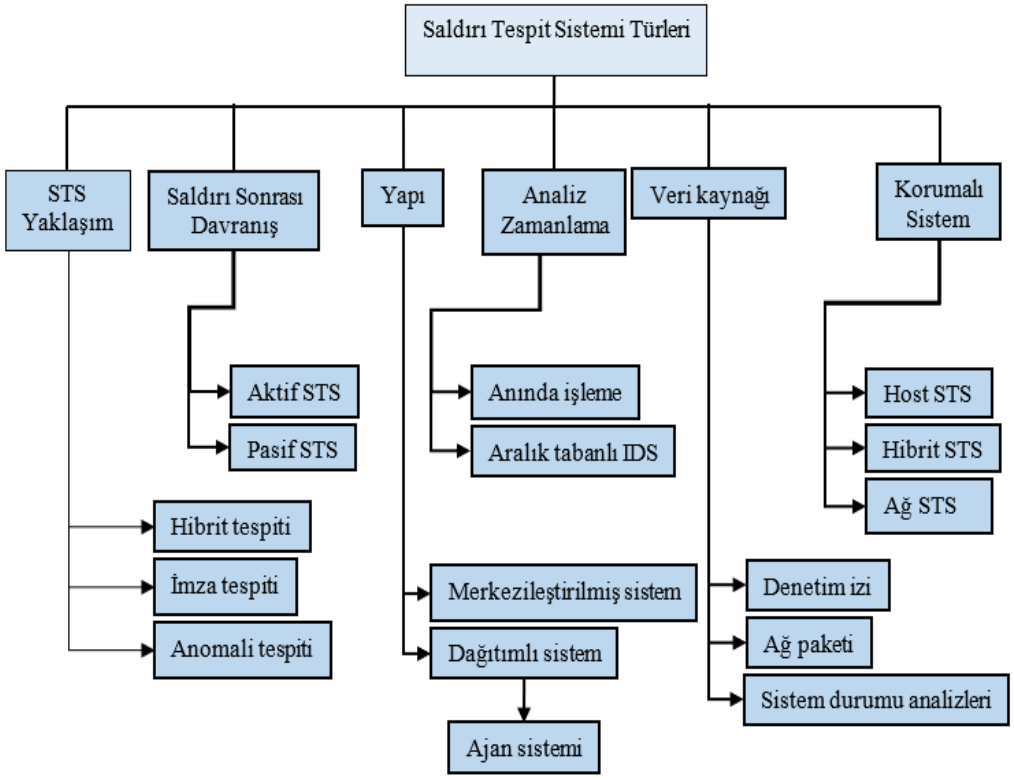
1983 yılında, Dr. Dorothy Denning ile SRI International arasında yeni bir saldırı tespit sisteminin geliştirilmesi amacıyla ortak proje üretilmiştir. Bu çalışmanın amacı ana bilgisayarlarından denetim kayıtlarını elde edip analiz ettikten sonra kullanıcı profili oluşturmaktır. Bir yıl sonra Dr. Denning ilk saldırı tespit sistemini sunmuş ama 1980 yılında ilk kavram ortaya atıldığı için James Anderson bu kavramın babası olarak kabul edilmiştir (Anderson, 1980).

Güvenlik sisteminin veri bütünlüğünü, gizliliğini ve kullanılabilirliğini ihlal eden her girişim kötü niyetli saldırı olarak tanımlanır, bu nedenle saldırı tespit sisteminin görevi, bu tür kaçak eylemleri düşürerek veya karartarak tespit etmek ve cevap vermektir. Saldırı tespit sisteminin genişletilebilirlik, uyarlanabilirlik ve doğruluk olmak üzere üç ana özelliği vardır. (Hari ve Aritra, 2012).

3.3. Saldırı Tespit Sistemi Kategorileri

Saldırı tespit sistemi, kötü amaçlı faaliyetleri yakalamak ve sistemi korumak için çalışır. Buna ek olarak, eylemin normal mi yoksa bir saldırı mı olduğunu fark etme yeteneğine sahiptir. Saldırgan, internet üzerinden sahte paketler göndererek veya yerel ağ ortamında yerleştirdiği virüsler yardımıyla bilgisayarın süreklilik sistemine zarar vermek için güvenlik ilkelerini ihlal etmeye çalışır. Bu tür tehditlerin üstesinden gelmek için saldırı tespit sistemleri yöneticiye bildiri göndererek süreci yönetmeye çalışır.

Farklı bakış açılarına göre gözlemlenen altı çeşit saldırı tespit yöntemi vardır. Bunlar; Korunan Sistem, Saldırı Tespiti (ID) yaklaşımı, Veri kaynağı, Analiz Zamanlaması, Saldırıdan Sonraki Davranış ve Yapı'dır (Alheeti, 2011).



Şekil.3.1 Saldırı tespit sisteminin altı farklı bakış açısından sınıflandırılması *(*Alheeti,2011)

3.3.1. Korumalı sistem yöntemleri

Korumalı sistem yöntemlerinde, Ana Bilgisayar (Host) saldırı tespit sistemi, birden fazla algoritmayı birleştirmek anlamına gelen Hibrit saldırı tespit sistemi ve Ağ saldırı tespit sistemi olmak üzere üç çeşit tespit modeli bulunmaktadır. Aşağıdaki bölümlerde bu üç türün nasıl çalıştığı ve özellikleri derinlemesine ele alınmıştır.

3.3.1.1. Ana Bilgisayar (Host) saldırı tespit sistemi

Host (Ana) bilgisayar saldırı tespit sistemi, sistem günlükleri, işlem hesaplama bilgileri ve dosya sistemleri gibi etkinlik kaynaklarını izleyerek ana bilgisayarları korumak için tasarlanmış bir araçtır. Bu model, bilgisayarlara gelen ve giden paketleri kontrol eden ve izleyen ana bilgisayarları tespit etmek için önemli bir role sahiptir. Tüm bilgileri topladıktan sonra, ana bilgisayar saldırı tespit sistemi, bilgisayar sisteminde şüpheli bir etkinliğin normal veya anormal olup olmadığını güvenlik uzmanı danışmanına gösterir ve bildirir. Buna ek olarak, ana bilgisayar saldırı tespit sistemi, yetkisiz kişilerin sistemdeki özel verilere erişmesine veya değiştirmesine izin vermemeye çalışır (Allen ve ark., 2000).

Host saldırı tespit sistemi, sistem çağrısı ve dışarıdan gelen tehditler gibi güvenliğin istikrarını ihlal eden şüpheli eylemlere odaklanır. Özellikle, sunucu sistemindeki değerli ve özel kritik sorunları korumak için kullanılır. Ayrıca Host saldırı tespit sistemi, ağ saldırı tespit sistemi ve Host saldırı tespit sistemi ayrı ayrı bilgisayarlara kurulduğunda ve ağ eylemleri olarak koruma sağlayacak biçimde yapılandırıldığında ağ saldırı tespit sistemi olarak işlev görebilir.

Host bilgisayar saldırı tespit sistemi, ana bilgisayara gelen zararlı saldırıları yakalamak için sunucularda ve iş istasyonlarında kurulan algılayıcıdan oluşmaktadır. Host saldırı tespit sistemi, işletim sisteminin yerel ayarlarını (denetim izleri) kullanarak ve günlük kayıtlarına erişerek sistem faaliyetlerini izleme ve düzenleme kapasitesine sahiptir. Bu sistem, hangi sunucu, protokol ve programın kullanıldığı gibi bazı önemli kaynakları toplayabilir ve görüntüleyebilir (Oktay ve Şahingöz, 2013).

3.3.1.2. Ağ saldırı tespit sistemi

Ağ saldırı tespit sistemi, belirli bir sunucudan gelen ağ trafiğini izlemek için işlev görür. Bu sistem, ağ anahtar cihazlarını belirleyen tüm paketlerin kaynaklarını ve hedeflerini kontrol eder. Buna ek olarak, bu sistem, yönlendiriciler gibi uç nokta cihazları tarafından tanımlanmış ağ trafiğine dayalı çeşitli koşulları önleme becerisine sahiptir. Ağ saldırı tespit sistemi, yeni paketi veri sisteminde depolanan önceki paketin imzası ile karşılaştırarak gelen paketin yetkilendirildiğinden emin olmak için ağdaki özel bir yerde kurulan belirli bir algılayıcıya sahiptir. Eğer paketin kötü amaçlı olduğu tespit edilirse, modül denetleyicisi paketi bırakır ve paketin bu nedenle atıldığından dolayı virüs içerdiğini gösteren bir mesaj bildirir. Ağ trafiği çeşitli katmanlardan oluşur ve her birinin verileri katmandan başka bir katmana aktarma veya iletme sorumluluğu vardır (Oktay ve Şahingöz, 2013).

3.3.1.3. Hibrit saldırı tespit sistemi

Hibrit saldırı tespit sistemi birden fazla algoritmayı bütünleştirmektedir. Başka bir deyişle, verimli ve daha fazla savunma sistemine sahip olmak için hem ana bilgisayar saldırı tespit sistemini hem de ağ saldırı tespit sistemini kullanmaktadır. Hibrit saldırı tespit sistemleri, bağımsız çalıştıkları için host saldırı tespit sistemi ve ağ saldırı tespit sistemi ile karşılaştırıldığında daha güvenlidir ve güçlüdür.

3.3.2. Saldırı tespiti yaklaşımları

Saldırı tespit yaklaşımları; yanlış kullanım ve aykırılık tespit sistemi olmak üzere iki ana kategoriye ayrılmaktadır. Yanlış kullanım (imza temelli ya da bilgi) modelde veriler kayıtlı olduğundan tüm saldırı imzalarını bilmektedir, ancak bu model yalnızca bilinen saldırganların imzalarını tespit edebilir, bilinmeyen imzaları tespit edebilme yeteneği yoktur. Aykırılık (davranış temelli) saldırı tespit yöntemi, ağdaki kullanıcı hareketlerine odaklanan istatistiksel davranışa dayanır.

3.3.3. Bir saldırıdan sonra saldırı tespit sisteminin davranışı

Saldırı sonrası saldırı tespit sisteminin davranış tipleri aktif ve pasif olmak üzere iki grupta incelenmektedir. Aktif olan yöntem, sistemi belirler ve belli IP adresindeki

trafiği engelleyerek otomatik cevap gönderir. Pasif saldırı tespit sistemi saldırı tehditlerini azaltmak ve tespit etmek için işlev görürken bilgi toplar ve yöneticiye geri gönderir. Yönetici, tehlikeli yolu engellemek için yapılandırmadan sorumludur. Pasif olan yöntem aktif olan yöntem gibi hemen tepki verme yeteneğine sahip değildir. Saldırı tespit sistemi, Şekil 1.1'de gösterildiği gibi Analiz Zamanlaması v.b. gibi diğer gruplara ayrılabilir.

3.3.4. Analiz zamanlaması

Bu tür saldırı tespit sistemi; anında işleme ve aralık tabanlı saldırı tespit sistemleri olmak üzere iki ana türden oluşan zamanlamaya dayanır. Saldırı tespit sisteminin anında işleme sürümü, davetsiz misafirlerden gelen herhangi bir olayı düzenli olarak yanıtlayarak çevrimiçi olarak çalışır, bu tür bir kimliğin kullanılması, prosedür sırasında meydana gelen büyük verileri kaydetmek için daha fazla RAM gerektirir. Ancak aralık tabanlı sürüm, hedefe erişmeden önce kötü amaçlı günlük dosyalarına sahip olup olmadıklarına bakılmaksızın paketlerin durumunu ve içeriğini kontrol etmesi anlamına gelen önceden tanımlı olarak adlandırılan özel bir işlem kullanır.

3.3.5. Saldırı tespit sistemi yapısı

Yapı tabanlı saldırı tespit sistemi, merkezi ve dağıtılmış olmak üzere iki grupta sınıflandırılır. Merkezi saldırı tespit sisteminde, veriler tek tek veya birden çok bilgisayardan toplanır, sonunda bu veri setleri kontrol için merkez konuma dönüştürülür (ARMD, Bro ve ARMD). Dağıtılmış saldırı tespit sisteminde ise, veriler her ana bilgisayarda ayrı ayrı toplanırken, analiz için her ana bilgisayarda dağıtılmış bir veri vardır. Bu saldırı tespit sistemine AAFİD (saldırı tespitinde otonom ajanlar) ve CSM (İş birliği güvenlik yöneticileri) örnek olarak gösterilebilir.

3.4. Saldırgan Türleri

Ağ ve ana bilgisayar saldırıları dört ana kategoride sınıflandırılabilir. Bunların her birinin hedefe saldırmak için kendi yöntemleri vardır. Aşağıdaki detay bilgiler sunulmuştur.

3.4.1. Hizmetin reddedilmesi (Dos)

DOS saldırısı, günümüzün en popüler ve en tehlikeli olanlarından biri olup, hızla artmaktadır. Bu saldırının, bir hizmetin Dağıtılmış Reddi (DDOS) olarak adlandırılan başka bir türü daha vardır. Bu iki tip arasındaki fark; DOS sadece hedefi kırmak için tek bir ana bilgisayar kullanır, DDOS ise zombiler veya botnetler adı verilen çoklu dağıtılmış bilgisayarlar kullanır. Bu ana bilgisayarlarda, sunucuyu meşgul eden ve yetkili kullanıcılar tarafından erişilemez hale getiren çok sayıda paket göndererek kurbanın sunucusuna saldırmak için kullanılabilecek virüsler ve istenmeyen kötü amaçlı yazılımlar vardır. Bu saldırıya örnek olarak sel (flooding) saldırısı verilebilir.

3.4.2. Kullanıcıdan Köke (U2R)

Bu tür, kullanıcı hesabını yönetici hesabına yükseltme olarak da belirtilebilir. Bu tip saldırılarda, halihazırda normal bir yetkiye sahip olan kullanıcılar önemli bilgilere ulaşmak için yönetici kullanıcı ayrıcalıklarına (sistemin köküne) erişmeye çalışır. Bu tür saldırılara örnek olarak Perl, xterm, attack overflow verilebilir.

3.4.3. Uzaktan yerele (R2L)

Bu tür saldırılarda, uzaktaki bir makineden farklı bir paket göndererek hedefin makinesine internet veya ağ üzerinden erişmeye çalışılır. Bu tür saldırganların Kullanıcıdan Köke gibi kullanıcı hesapları yoktur. Amaçları önemli bilgileri çalmak ve sistemin işlevselliğine zarar vermektir.

3.4.4. Probe

Saldırganların kullandığı son yöntem hem makineyi hem de ağı izleyen Probe olarak adlandırılır, bu tür saldırganların amacı sistemin güvenliğini ve sürekliliğini ihlal etmektir. Başlıca güçlü yanı, saldırı meydana gelirken ortaya çıkan tüm bilgileri eksizsiz toplamak için hedef sistemin değişkenliğini taramaktır. Bağlantı noktası taraması, bu saldırı türünün bir örneğidir. Bu yöntem kısmen veri madenciliğinde kullanılır (Örneğin bağlantı noktası taraması(port sweep), mscan vb.) (Paliwal and Ravindra, 2012)

3.5. Makine Öğrenimi

Tom M. Mitchell, “Makine Öğrenmesi, 1 Mart 1997” adlı kitabında makine öğrenimi kavramını açık bir şekilde tanımlamıştır. Temelde “Deneyimle otomatik olarak kendini geliştiren bilgisayar programları oluşturmanın bir yolu var mı ve öğrenme sürecini kontrol eden temel kurallar nelerdir?” sorularına yanıt aramıştır (Mitchell, 1997).

Bu soru, tasarım ve öğrenme görevlerinin birçok önemli kısmına, örneğin deneyimlerine dayanarak öğrenebilen bir sistemin nasıl oluşturulacağına odaklanmaktadır. Bu arada, günümüzde makine öğrenimi algoritmalarına (Machine learning (ML)) dayalı çalışan birçok uygulama, davetsiz misafirleri engelleyen veri madenciliği programları, sistem olaylarını sınıflandırabilen ve filtreleyebilen bilgi güvenliği sistemleri veya hatta otonum otomobil üreten mekanik üreticiler gibi çeşitli alanlarda geliştirilmektedir. Temel makine öğrenme algoritmaları; yapay zekâ, olasılık, istatistik, felsefe, biyoloji, hesaplama karmaşıklığı, bilgi teorisi, bilişsel bilim gibi farklı alanlardan esinlenilerek türetilmiştir.

Alpaydın (2010) “Makine öğrenmesi nedir?” sorusunu şöyle tanımlamıştır. Makine öğrenimi, daha önceki bir deneyimi göz önünde bulundurarak veya veri kümesini kullanarak bir performansı optimize etme işlevi gören bilgisayar programıdır. Başka bir deyişle, makine öğrenme algoritması, bilgisayarı doğrudan kodlamadan bir problemi çözmektedir, ancak bunu yapmak için sistemin verilere veya geçmiş deneyimlere sahip olması gereklidir (Alpaydın, 2010).

Makine öğrenimi, verileri öğrenmek için en azından istatistiksel çözüm gerektiren, daha sonra karmaşık görevleri düzenleyen ve kesin tahmin çözümlerini ortaya koyan verimli bir sistemin yapılandırılmasına dahil olan yapay zekanın ana merkezidir. Günümüzde makine öğrenimi robotlar, örüntü tanıma, optimizasyon ve öğrenme araştırması; tıp, yazılım ve geleneksel iş problemlerinde uygulamalar kullandı. Özellikle iş problemi alanında, veri bilimi, veri madenciliği gibi birçok alanda yer almaktadır (Patrick ve ark., 2014).

3.6. Makine Öğrenme Yöntemlerinin Çeşitleri

Alpaydın (2010) kitabında beş farklı makine öğrenme yönteminden bahseder. Bunlar; Birliktelik Kuralı Yöntemi, Sınıflandırma, Regresyon, Denetlimli ve

Denetimsiz Yöntemler ve Takviyeli öğrenme yöntemleridir. Bu yöntemlerin her biri aşağıda ayrıntılı olarak açıklanmıştır.

3.6.1. Birliktelik Kuralı Yöntemi

Birliktelik Kuralı yöntemi iki nesne arasındaki ilişkiyi elde etme prensibine dayanan bir makine öğrenme yöntemidir. Örnek olarak müşteriler tarafından satın alınan ürünler arasındaki ilişkiyi elde etmek için süpermarketlerde görebileceğiniz birliktelik analizi uygulaması olan sepet analizi uygulası verilebilir.

3.6.2. Sınıflandırma

Sınıflandırma, makinenin öğrenmesinin ikinci yöntemidir. Son sonuçlarını tahmin etmek için iki veya daha fazla sınıfı sınıflandırır. Tez kapsamında uyguladığımız veri kümelerini ele alırsak (NUSWNB15 ve NSL-KDD) bu iki veri kümesinin etiketi normal veya anormal örneklerden oluşur. Sınıflandırma algoritması kullanılarak sınıflandırılmak istenildiğine, sonuç normal ya da kötü amaçlı olarak ortaya çıkacaktır.

3.6.3. Regresyon

Regresyon, makine öğreniminin üçüncü yöntemidir. Bir evin fiyatını tahmin edebilen bir sistemimiz olduğunu varsayalım. Evin oda sayısı, yapıldığı yıl, modern olup olmaması gibi özellikleri bize evin değerini verecektir. Bu problem regresyon problemi olarak adlandırılır. Bu bağlamda sınıflandırma ve regresyon denetimli öğrenme yöntemleridir.

3.6.4. Takviyeli Öğrenme

Takviyeli öğrenme (Reinforcement Learning), bulunduğu ortamı algılayan ve kendi başına kararlar alabilen bir sistemin, hedefine ulaşabilmesinde doğru kararlar almayı nasıl öğrenebileceğini gösterir. Bu yöntem robotik, oyun programlama, hastalık teşhisi ve fabrika otomasyonu gibi alanlarda sıklıkla kullanılır.

Takviyeli öğrenmede bir eğitmen bulunur fakat denetimli öğrenmedeki gibi sisteme çok detay vermez veya veremez. Bunun yerine öğrenen sistem bir karar

verdiğinde bu kararın doğru olduğu durumlar için sistemi ödüllendirir ve yanlışlar için de cezalandırır. Amaç, öğrenen sistemin denediği olası durumların hedef olup olmadığının kontrolü ve denenen doğru veya yanlış tüm durumların hatırlanmasıdır.

3.6.5. Denetimli Öğrenme

Denetimli öğrenme algoritmaları, bir işlev oluşturmak için eğitim verilerinin kullanılmasına bağlıdır. Ayrıca eğitim verileri, sınıf etiketi adı verilen giriş ve çıkış vektörlerinden oluşur. Öğrenme algoritması, bir sınıflandırıcı oluşturmak için giriş ve çıkış örnekleri arasındaki tahmini belirlemek için kullanılır. Model geliştirildikten sonra bilinmeyen örnekleri öğrenilmiş sınıf etiketlerine ayırabilir. Özgün denetimli makine öğrenimi sınıflandırıcısının kullanılması izinsiz giriş tespiti algılama zorluklarının üstesinden gelebilir. Bazı makine öğrenmesi yaklaşımları şunlardır; SVM, YSA, DT, SOM, NBC vb. (Fong ve ark., 2009)**.**

3.6.6. Denetimsiz Öğrenme

Denetimsiz (kümeleme) Öğrenme etiketlenmez, ancak K-Means Kümeleme algoritması gibi denetimsiz algoritmalar verileri türlerine göre gruplandırır. Tez kapsamında kullanılan veri kümeleri de bu öğrenme modeline uygundur. Denetlenen öğrenme verileri, normal veya anormal olarak etiketlenir. Modelin öğrenme süreci, normal ve kötü amaçlı olmak üzere girdi veri etiketlerini anlamalı ve kategorize etmelidir (Tsai ve Ark., 2009).

3.7. Denetimli Öğrenme Algoritmaları

Tezde önerilen modelde denetimli sınıflandırma algoritması tercih edildiği için bu bölümde bazı denetimli algoritma makinesi öğrenme yaklaşımları ele alınmıştır (SVM, RF, KNN, DT ve NBC).

3.7.1. Destek Vektör Makinası (SVM)

Destek vektör makinesi (SVM) kavramı Vapnik adlı bir araştırmacı tarafından ortaya atılmıştır(Bahareh ve ark., 2014). Algoritmanın çalışma prensibi şu şekildedir;

başlangıçta giriş vektörü daha yüksek boyutlu özellik haritasına dönüştürülür, ikinci olarak, boyutsal uzayda en uygun ayrık hiper düzlem elde edilir. Ek olarak, destek vektörü tüm eğitim örneklerinden ziyade hem karar sınırına hem de hiper-düzlemi ayırmaya karar verir. Bu şekilde aykırı değerler için de daha güçlü sonuç elde edilmiş olur.

Özellikle, destek vektör makinesi (SVM) sınıflandırıcı algoritması, iki çeşit sınıfa sahip eğitim veri setini ayırmak için görev yapan ikili sınıflandırma için önerilmiştir. Karar sınırına yakın veya etrafındaki eğitim veri örneklerine destek vektörü denir. Algoritmanın doğrusal, polinom, radyal temel fonksiyonu olmak üzere üç yöntemi vardır. Veriler kolayca ayrılabilir ise doğrusal yöntem kullanılır, aksi takdirde diğer yöntemlere yer verilir. Ceza parametresi faktörü, kullanıcının yanlışlıkla sınıflandırılan özellikler ile genişlik karar sınırı arasında değiş tokuş yapmasına olanak tanımaktadır. (Bahareh ve ark., 2014).

Destek Vektör Makinesi, denetimli makine öğrenme algoritmalarının en popülerlerinden biridir. Daha çok sınıflandırma ve regresyon problemlerinin çözümünde kullanılmaktadır. Başarılı performansı nedeniyle sınıflandırma algoritmalarında üst sıralarda yer almaktadır. Boyutsal uzay veri örneklerini çizmek için haritalandırılır, sonrasında iki sınıfı ayıran hiper düzlem elde etmek için sınıflandırma işlemi başlatılır. Sınırlara yakın olan örnekler destek vektörü olarak tanımlanır. SVM eğitim örneklerinin, destek vektörleri olarak adlandırılan farklı alt gruplara bölünmesi durumunda, karar fonksiyonu bu destek vektörleri tarafından belirlenir. (Belavagi ve Balachandra, 2016).

Destek Vektör Makinesi (SVM), eğitim veri kümesi vektörlerini her örneğin sınıf etiketine göre boyutsal alanda çizen denetimli öğrenme modelidir. SVM, sınıflandırma problemini kuadratik optimizasyon problemi olarak kabul eder. SVM, destek vektörlerini tanımlayarak veri örneklerini sınıflandırır. Destek vektörleri, hiper düzlemine yakın olan eğitilmiş veri kümesinin bir parçasıdır. SVM'ler, bir çekirdek işlevi kullanılarak hiper düzlemin yüzeyini verilere sığdırmak için genel bir mekanizma sağlar. Kullanıcı, destek vektörünü seçmek için eğitim işlemi boyunca SVM için polinom, sigmoid veya doğrusal fonksiyonlar kullanabilir. SVM parametresi, veri noktası ve hiper düzlem arasındaki ayrım çizgisine bağlıdır. SVM'yi kullanmanın faydalarından biri, aşırı öğrenmeyi önlemek için veri setinin özelliklerinin sayısını en aza indirmeye ihtiyaç duymamasıdır, bu saldırı tespiti gibi alanlarda önemli bir fayda

sağlar. SVM'nin bir başka faydası ise küçük olasılık hatalarını önceden tahmin etmektir (Mukkamala ve ark., 2002).

İki nedenden dolayı saldırı tespiti için sınıflandırıcı olarak SVM'yi uygulamak önemli olmuştur; Birinci neden, etkin hızının olmasıdır. Algoritma hızı arttıkça, keşif ortamlarında o kadar çok tercih edilir. Hız performansı, gerçek zamanlı kötü niyetli saldırıları tespit etmek için çok önemlidir ve bir anahtardır. İkinci neden ise ölçeklenebilirliktir. SVM, her iki sınıflandırma zorluğunun alan boyutuyla ilişkili olmadığı kadar veri örneklerinin miktarını da dikkate almaz. Bu da, algoritmanın çok sayıda örneği öğrenme yeteneğine sahip olduğu anlamına gelir. Ayrıca SVM'nin ölçeklendirilmesi yapay sinir ağlarına göre daha elverişlidir (Mukkamala ve ark., 2002).

Destek vektör makinesinin doğrusal ve doğrusal olmayan sınıflandırıcı olmak üzere iki sınıflandırıcı tekniği vardır. Bunlardan ilki doğrusal ayırıclarıdır.

Özellik uzayında çizilen eğitim örnekleri doğrusal olarak ayırt edilebilir olduğunda kullanılması önemlidir. Bu, iki sınıfa ayrılan hiper düzlem çizgisinin varlığı anlamına gelir. İki sınıftan oluşan ikili sınıflandırma problemine dayanan denetimli bir problemimiz olduğunu ve yine eğitim veri setinin özellik uzayının, boyutsal uzay üzerinde M vektörleri içerdiğini varsayalım. Karar fonksiyonu bu ifadeye $(sgn[f(x)])$ dayanmaktadır, burada $f(x)$ hiper düzlem ile ilişkili bir ayırt edici fonksiyonu temsil etmektedir ve fonksiyon tanımı aşağıdaki gibidir; (Farid, ve ark(2004)

$$f(x) = w.x + b \tag{3.1}$$

Bu tür bir hiper düzlem elde etmek için W ve b $y_i = (w.x + b) > 0$ $(i = 1,2,3 \dots M)$ fonksiyonu kullanılarak öngörülür.

İkincisi, (*Non-linear (kernel)* doğrusal olmayan (çekirdek)) ayırıcıdır. Bu ayırıcı, veri vektörleri harita uzayında doğrusal olarak bölünemediğinde kullanılmaktadır. Problemin üstesinden gelmek için, doğrusal alandan çok daha geniş boyutlu alan içeren bir doğrusal olmayan ya da çekirdek yöntemi önerilmiştir. Çekirdek hiper düzlemini çizmek için çekirdek fonksiyonlarından biri kullanılır. Çekirdek fonksiyonları şunlardır;

- *Polinom (homojen)*
- *Polinom (homojen olmayan)*
- *Radyal Temel Fonksiyon (RBF).*

Polinom (homojen) çekirdeği, denklem 3.2'i kullanılarak ifade edilebilir.

$$K(\vec{X_i}, \vec{Y_i}) = (\vec{X_i} \cdot \vec{Y_i})^d \quad (3.2)$$

$K(\vec{X_i}, \vec{Y_i})$ Çekirdek fonksiyonudur, d polinomun derecesini, X_i ve Y_i uzaysal haritadaki vektörleri temsil etmektedir. Denklem 3.2'de yer alan fonksiyona polinom (homojen olmayan) çekirdek için özelliklerin entegrasyonunu etkileyen ek bir sabit (serbest parametre) değişken eklenir. Denklem 3.3'deki gibi tanımlanır.

$$k(y, x) = (X^T y + c)^d \quad (3.3)$$

Üçüncü çekirdek fonksiyonu ise, karesel uzaklık matrisi için Öklid uzaklığını kullanan Radyal Temel fonksiyonudur (RBF). Ayrıca bu fonksiyon diğer çekirdek fonksiyonları arasında en çok kullanılanlardan biridir. Fonksiyonun yapısı denklem 3.4'de gösterilmiştir.

$$k(x, x') = \exp\left(-\frac{(||x-x'||)^2}{2\sigma^2}\right) \quad (3.4)$$

Burada σ, sabit veya serbest bir parametredir.

SVM; doğrusal SVM sınıflayıcı (LSVM) ve Doğrusal Olmayan SVM sınıflandırıcı olarak iki tür sınıflandırıcıya bölünmüştür. Veriler ayrılabilir olduğunda, LSVM kullanılabilir. LSVM'nin görevi, verileri iki sınıfa ayırarak verileri eğitmektir. Denklem 3.5 de fonksiyon verilmiştir (Babaoğlu ve ark., 2010).

$$\begin{aligned} &(W.X_i) + W_0 \geq +1 - \xi_i, if\ Y_i = +1 \\ &(W.X_i) + W_0 \geq +1 - \xi_i, if\ Y_i = -1 \\ &\text{Yada} \\ &[(W.X_i) + W_0] \geq, i = 1, \ldots, n \end{aligned} \quad (3.5)$$

Burada ξ_i, küçük hatalara sahip bazı verilerin kabulünü sağlayan bir değişkendir. Veri kümesi doğru şekilde sınıflandırılırsa, ξ_i'e ihtiyaç duyulmaz. Diğerleri arasında en iyi hiper düzlemi elde etmek için, bu fonksiyon denklem 3.6'daki gibi indirgenerek kullanılır.

$$A\sum_{i=1}^{n}\xi_i + 1/2||W| = ||W||^2 \quad (3.6)$$

A, sınıflandırma ve karmaşıklık performansı arasındaki değişimin bir parametresidir. Destek vektör makinesi ile hiper düzlem arasındaki mesafeyi maksimize etmek, optimum hiper düzlem seçilmesine bağlıdır.

Doğrusal olmayan SVM sınıflandırıcısı, hiper düzlem kullanarak verileri ayırmak mümkün olmadığında kullanılabilir. Doğrusal olmayan durum (ayrıca giriş verilerinin uzay haritası) çekirdek işlevi aracılığıyla daha yüksek boyuta dönüştüğünde**,** çekirdek fonksiyonları denklem 3.7'de yer alan şekliyle tanımlanır (Babaoğlu ve ark., 2010)

$$K(x,\bar{x}) = (\ \Phi(x).\Phi(\bar{x})\ ) = \ \Phi(x)\Phi(\bar{x}) \quad (3.7)$$

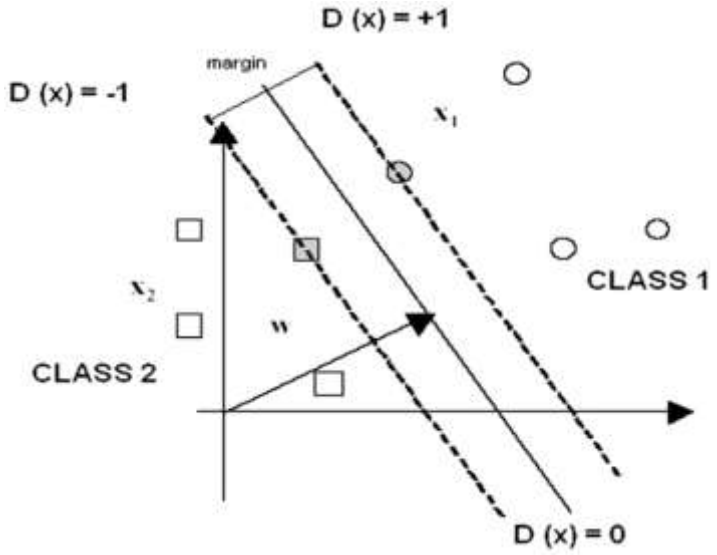
Çekirdek işlevleri, Mercer'in durumunu karşılamalıdır. Böylelikle, 3.8'deki formüllere SVM'nin ikinci dereceden optimizasyon probleminin amacı ve karar fonksiyonu dönüştürülmüştür.

$$Maximum\ =\ W(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\ \sum_{i,k=1}^{n} a_i y_i\, a_k a_i y_k K(x_i, x_k)$$

$$f(x)\ =\ W(a) = sign\left(\sum_{i=1}^{n} a_i - \frac{1}{2}\ \sum_{i,k=1}^{n} a_i y_i\, a_k a_i y_k K(x_i, x_k)\right) \quad (3.8)$$

Sıklıkla kullanılan çekirdek fonksiyonları şunlardır;

- Dod Çekirdek işlevi: $K(x,\bar{x}) = x.\bar{x}$
- Polinom Çekirdeği işlevi: $K(x,\bar{x}) = (x.\bar{x} + 1)^d$, d çekirdek derecesidir.
- Radius temel fonksiyon çekirdeği (RBF). $k(x,x') = \exp(-\frac{(||x-x'||)^2}{2\sigma^2})$, σ gerçek bir pozitif sayıdır.



Şekil.3.2 SVM Yapısı (Babaoğlu ve Ark., 2010)

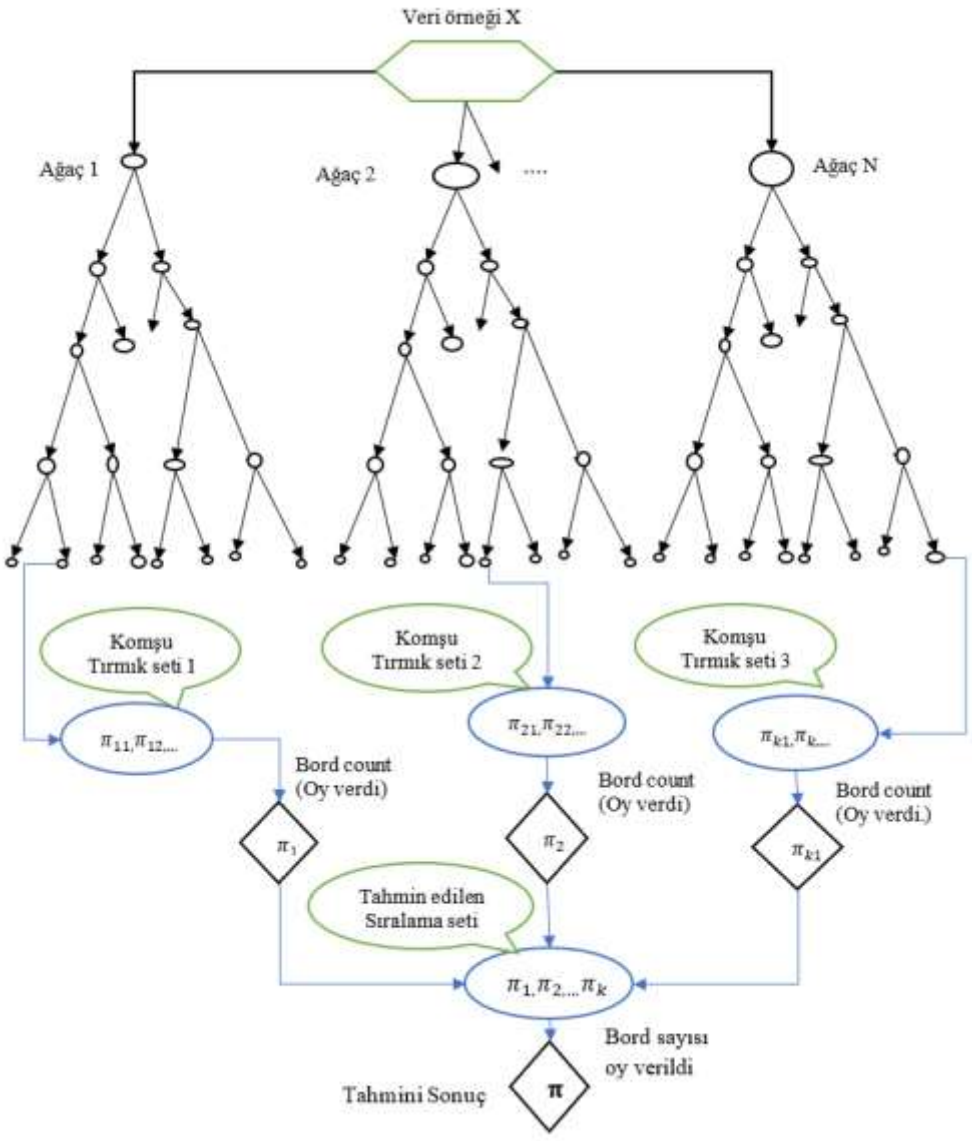
3.7.2. Rastgele Orman algoritması (RF)

Karar ağaçlarının yetersizliğini gidermek için rastgele orman algoritması (Random Forest (RF)) Breiman tarafından sunulmuştur. Özellikle belirsizlik karmaşası, öğrenilen veri kayıtlarında ortaya çıkar. RF oluşturmak için, çeşitli sınıf tahminlerini elde etmek için çok sayıda rastgele karar ağacı oluşturulur ve bunların arasında en çok oy alan sınıf seçilir. Ayrıca, RF her karar ağacı için bağımsız kayıtlar seçerek karar ağaçları arasındaki ilişkiyi azaltır. Bu yöntem, sınıflandırma, regresyon ve hayatta kalma analizi gibi birçok bilimsel bölümden yararlanıldığı için bazı makine öğrenme algoritmalarına kıyasla çok daha güçlü ve verimlidir. RF algoritmasını belirlemek için yordayıcıların sayısı, ormandaki yetişkin ağaçların sayısı, ayırma kuralları olmak üzere üç parametreye ihtiyaç duyulur (Utkina ve ark., 2019).

Rastgele orman (RF), potansiyel bir öğrenmeye ve iyi sonuçlara sahip olan denetimli bir makine öğrenme algoritmasıdır. Algoritma 2001 yılında Breiman tarafından önerilmiştir. Karar Ağaçlarının sayısını toplar ve tahmin sonucu en çok oy alan sınıfı seçer. Model, başarılıdır ve farklı amaçlarla da kullanılmıştır. Rastgele orman algoritmasını oluşturmak için, farklı eğitim veri setlerine (D_i) sahip karar ağaçları oluşturmak gerekir. Burada *i*, 1 den karar ağacı sayısına kadar değer almaktadır. Her bağımsız veri, ayrı ayrı karar ağaçlarını eğitmek için kullanılır. Buna ek olarak, bu veri kümeleri orjinal veri kümesi D'den rasgele bir şekilde seçilir. Her karar ağacı bu rastgele ve farklı şekilde oluşturulmuş eğitim veri setlerinde dallanır. Özellikle, rasgele

oluşturulmuş veri kümelerinde her karar ağacının kök düğümü belirlenirken, en yüksek bilgi kazanımı hesaplanılır. Bilgi kazanımı IG ile gösterilir. $N_s = [\log 2 \ A + 1$ Denkleminde, N_s küme sayısını, A ise rasgele seçilen orjinal özellikleri gösterir. Genellikle, oluşturulan tekil karar ağacı, ancak durma durumuna gelirse algoritma durdurulur. Ormanda oluşturulan tüm ağaçaların budanmasını izin verilmez. Eşiklere göre tek boyutlu öznitelik değerini dikkate alan karar ağaçları oluşturmak için eğitim verileri (D) yinelemeli bir şekilde bölümlere ayrılır. (Yangming ve ark., 2018). Her düğümdeki veriler için denklem 3.9 kullanılır.

$$\begin{cases} a^i \geq a^i_{Threshold} & go\ left\ child \\ otherwis & go\ right\ child \end{cases} \tag{3.9}$$



Şekil.3.3 Etiket sıralaması için rastgele ormanın şematik bir gösterimi (Yangming ve ark., 2018)

3.7.3. K en yakın komşuluk algoritması

K-en yakın komşular (KNN), sınıflandırma ve regresyon için kullanılan, parametrik olmayan ve benzerlik tabanlı denetimli öğrenme algoritmasıdır. KNN sınıflandırma algoritmasının kuralı, örneklere en yakın k komşuyu algılamak ve her eğitim verisini uygun sınıfa göre belirlemektir. Birkaç nokta arasındaki mesafeyi hesaplamak için algoritma Öklid mesafesini kullanır. K parametresi vardır veya bazen algoritmada manipüle edilmiş pozitif tam sayı değer alan komşu sayısını temsil eden hiper düzlem olarak adlandırılır. Ayrıca, k parametresinin değerinin bire eşdeğer olduğu durumlarda, verinin örneği birinci sınıfta yer alacaktır. Aksi halde sınıf, en yakın komşunun çoğunluk oyuna göre belirlenir (Buttrey ve Karo, 2002).

$$(U_i . Z_y) = [\sum_{n=1}^{a} C_n (U_{in} - Z_{yn})]^{1/2} \quad (3.10)$$

Justin ve Mark (2018), KNN'i hem regresyon hem de sınıflandırma amacıyla sıklıkla kullanılan makine öğrenme algoritmalarından biri olarak açıklamışlardır. KNN kavramsal olarak, ölçülen benzer değere sahip farklı örnekleri olan bir varsayım olarak ifade edilir. Öklid uzaklık formülü kullanılarak örnekler arası mesafe hesaplanarak aynı değer aralığına sahip örnekler elde edilir (Lee ve Styczynski, 2018).

$$d(A, B) = \sqrt{(x1 - x2)^2 + (y1 - y)^2} \quad (3.11)$$

3.7.4. Karar Ağaçları

Paez ve ark. (2018) tarafından tanımlanan Karar Ağacı (Decision Tree (DT)), hem sınıflandırma hem de regresyon amaçları için kullanılan, makine öğrenme algoritmalarından biridir. Bağımsız veri kümelerini eğitmek için bir karar ağacı algoritması gerçekleşir, bu yordam, süreç sona erene kadar yinelemeli olarak devam eder. Karar ağacının kökü ve düğümleri vardır. Karar ağacını uygulamak için yukarıdan aşağı açgözlü arama kullanan ID3 ismi verilen özel ve birincil algoritma kullanmak çok önemlidir. ID3, karar ağacını oluşturmak için bilgi kazancı ve entropi kullanır. Öznitelikler farklıysa entropi bir, aksi takdirde sıfır değerini alır. Karar ağacını oluşturmak için iki tür entropi hesaplanır. Bunlar; bir özniteliğin sıklık tablosu ve iki

özniteliğin sıklık tablosudur. Bilgi kazanımının işlevselliği, veriler örneğe ayrıldığında azalmaya başlar. Karar ağacı, en yüksek bilgi kazancına sahip olan özelliği seçer. (Quinlan, 1986).

En çok kullanılan makine öğrenme algoritmalarından biri karar ağacıdır. Üstesinden gelinmesi gereken önemli bir şey, hem eğitimden hem de test veri setinden bilinmeyen değerlere sahip özellikten nasıl yararlanılacağıdır. Karar ağacının her düğümü, bir örneğin testini temsil eder ve yaprak düğümler sınıflandırmayı temsil eder. Test sınıflandırmasının başlangıç noktası yukarı köktür ve sınıflandırmanın elde edilen niteliğini gösteren yaprak düğümüne yaklaşana kadar aşağı iner. Doğruluğu arttırmak için bilinmeyen değerlerin eğitim verilerindeki etkisini göstermek için birçok yöntem sunulmuştur. (Paeza ve ark., 2018).

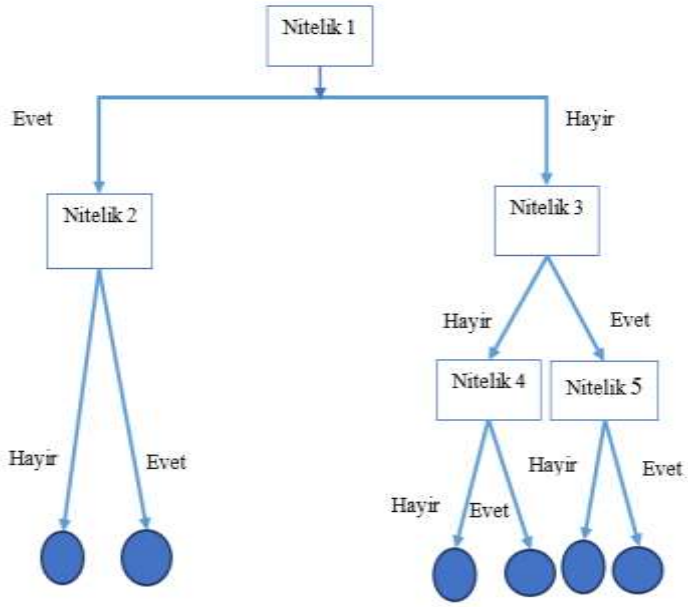
Karar Ağacı, genellikle sınıflandırma için parametrik olmayan makine algoritmasının kullandığı, nitelikleri uygun sınıflarına göre sınıflandırmaya çalışan denetimli bir sistemdir. Bu yöntemin veri madenciliğinde kullanımı sıklıkla görülmektedir. Ayrıca, karar ağaçları, kök, yaprak ve uç noktalardan oluşan yukarıdan aşağıya bir yapı biçiminde sınıflandırılmış veri kümesi oluşturur. Karar ağacı oluşturmak için ID3 algoritmasının kullanılması şarttır.

ID3, örneklerin homojen alt kümesini hesaplamak için entropi ve bilgi kazancını kullanır ve verileri bölmek veya ayırmak için karar ağacının karar verme yöntemini kontrol eder. Alt küme verileri birbirine benziyorsa, entropi sıfırdır ve alt küme verileri bölünebilir olduğunda entropi birdir. Karar ağacı oluşturmak için frekans tablosu kullanılarak iki tür entropi hesaplanır, birincisi bir özelliğin frekans tablosunu kullanan entropi ve ikincisi iki tablonun frekans tablosunu kullanan entropidir. Karar ağaçları, görselleştirilebildiğinden ayırt etmenin ve yorumlamanın çok kolay olması gibi önemli bir avantaja sahiptir. Karar ağaçları hem sayısal veri hem de kategorik veriler için kullanılabilir. Bilgi kazancı (Information gain (IG)), karar ağacı düğümünü oluşturmak için her bir özelliğin entropisi hesaplandıktan sonra diğer özellikler arasında en yüksek özellik değerinin seçilmesinden sorumlu olan ve karar veren yöntemdir. Öte yandan, bir karar ağacı oluşturmak, diğerleri arasında en yüksek bilgi kazanımına sahip olan özelliği seçmektir. Denklem 3.12 de entropinin hesaplanması gösterilmektedir.

$$\text{Enropy} = -\text{N}\log_2 \text{N} - \text{M}\log_2 \text{M} \tag{3.12}$$

Diğerleri arasında en uygun ve en yüksek özelliği seçmek için, Bilgi kazanımı, öngörülebilir entropiyi ve yeni hesaplanmış entropiyi çıkararak karar verir, bu prosedür tüm özelliklere uygulanır ve nihayetinde bilgi kazancı en yüksek olan özellik seçilir, ayrıca bu özellik sınıfın kök düğümü olur. Denklem 3.13'de bilgi kazanma denklemi gösterilmiştir.

$$\text{Informaiton Gain}(X, Y) = \text{Entropy}(X) - \text{Entropy}(X, Y) \tag{3.13}$$



Şekil 3.4. Prototipik Karar Ağacı (Antonio ve ark., 2018)

3.7.5. Naif Bayes Sınıflandırıcısı

Naif Bayes sınıflandırıcısı, bağımsızlık nitelikleri (varsayımlar) üzerinde büyük ölçüde işlev gören basit bir Bayes olasılık teoremidir. Her özelliğin başkalarını etkilemeden kendi olasılığı vardır. Bununla birlikte, naif sınıflandırıcıların sonucu diğer algoritmalara kıyasla çoğunlukla daha doğru ve daha etkili olmaktadır.

Naif Bayes sınıflandırıcılarının hangi koşullar altında yetkin bir şekilde çalıştığını gösteren deneysel bir çalışma mevcuttur (Saurabh ve Sharma, 2014) ve buna göre yetkinliği arkasındakisebep, saf Bayes sınıflandırıcısının hata sonucunun üç bileşene neden olmasıdır. Bunlar; varyans, eğitim veri kümesinin gürültüsü ve biaslardır. Makine öğrenmesi algoritmalarını kullanarak eğitim verilerinin gürültüsü azaltılabilir ve eğitim verilerinin farklı sınıflara ayrılması sağlanabilir. Veri kümeleri

gruplandırılmaları için çok büyük hale geldiğinde, bu bias hatası olarak adlandırılan bir hataya neden olur, bunun aksine eğitim verileri çok küçükken varyans adlı bir hata görünür. (Saurabh ve Sharma, 2014).

Naif Bayes sınıflandırıcısı, diğer sınıflandırma algoritmalarının aksine Bayes teoreminin temel alındığı sınıflandırma olarak uygulanabilecek karmaşık olmayan denetimli bir modeldir. Sadece sınıflandırma olarak değil, aynı zamanda tahmin edebilme yeteneğine sahip yöntemlerden biridir. Naif Bayes sınıflandırıcısı, bağımsız özellikleri yönetmek için Bayes teoremine dayanan bilgi toplama modelini çalıştırır (Amjad ve ark., 2018).

Naif Bayes, sadece veri madenciliğinde değil, aynı zamanda makine öğreniminde de iyi bilinen bir algoritmadır. Bir olayın olasılığını tahmin eder ve nitelikler birbirinden bağımsızdır. Bayes teoreminden basit ve etkili bir yöntemdir, tüm özelliklerin (niteliklerin) birbirini bağımsız olduğu varsayımı, modelin oluşturulmasını kolay hale getirir. İzinsiz giriş tespiti ve diğer birçok alanda uygulanmıştır (Levent ve ark., 2012).

Naif Bayes, geleneksel olmayan özellikler tarafından elde edilen sonuç olasılığına odaklanan olasılık temelli bir yöntemdir. Model kolay oluşturulur, hesaplama süresinde daha az zaman alır ve verimlidir. Her bir özelliğin olasılık sonuçları bir diğerinden farklıdır. Üstelik çoğu zaman başarılı sonuçlar verir.

Naif sınıflandırıcı, eğitim aşamasında belirli bir özelliğin olasılıklarını hesaplar ve ardından bu olasılığı kaydeder. Bu işlem her özellik için yinelenir, bu, nitelikler arasındaki uygun olasılığı değerlendirmek ve elde etmek için birkaç defa tekrarlanır. Test aşamasında, en kötü duruma göre verilen her bir özelliğin olasılığını hesaplamak için harcanan zaman X özellik sayısı ile karşılaştırılır. Bununla birlikte hem eğitimde hem de en kötü test aşamasında harcanan zaman aynıdır. (Panda ve Patra, 2007).

Naif bayes çeşitli alanlarda kullanılmaktadır. Naif Bayes kullanmanın temel kavramı, iki rasgele olaya sahip olan sonlu olasılıkları hesaplamaktır. Pek çok yazar, araştırmadaki saf Bayes sınıflandırıcısını basit ve özniteliklerinin bağımsız varsayım olması nedeniyle sınıflandırma modeli olarak uygulamıştır, bu da, her değişkenin diğer özniteliklerin olasılığını etkilemeden kendi olasılığına sahip olduğu anlamına gelir. Buna ek olarak, model, diğer örnekler arasında sınıftaki en yüksek örnekleme olasılığını tahmin eder.

Naif Bayes algoritması makine öğrenmesi algoritmaları arasındaki en iyi sınıflandırma algoritması olmasa da, neredeyse varolan algoritmaların hepsini

karşılatırsak bile daha iyi bir çözüm sunar. Basitliği nedeniyle sınıflandırıcı olarak spam filtreleri, kanser teşhisi ve yüz tanıma vb. gibi çok sayıda uygulamada yer almıştır. Naif Bayes algoritması doğru sonuçları öngörmek için etkili ve azaltılmış bir eğitim verisi gerektiren başarılı sonuçlar verir. Tahmin sürecinde sınıflandırıcı, sınıfını belirleyerek verilen hedef örneğin olasılığını tahmin etmek için eğitim verilerini alır (Chong-zhive ark., 2018).

Naive Bayes sınıflandırıcı, özellikleri birbirinden bağımsız belirli bir verinin sınıfına göre olasılığını tahmin eden denetimli bir makine öğrenme algoritmasıdır. Ayrıca bağımsızlık, tahminin güçlü olmamasına neden olur, pratikte Naif Bayes algoritması sınıflandırıcı daha gelişmiş sınıflandırıcılarla daha iyi sonuçlar elde edebilir. Bu yöntem, anormalin 1'e ve normalin 0'a atadığı iki sınıf optimizasyonunda çok önemli bir rol oynar. (Rish, 2014).

Naif bayes sınıflandırıcısı, çeşitli sınıflandırma problemlerinin üstesinden gelmek için literatürde sıklıkla kullanılan bir yöntemdir. Görevin modeli, verilere ait sınıfın olasılığını tahmin etmektir. Buna ek olarak, nitelikler birbirinden bağımsız olan özel bir karaktere sahiptir. Model denetlenmesine rağmen, sınıf etiketleri bilinmektedir. Her bir özniteliğin olasılığı elde edilen sonuca benzer şekilde etkilenir, bu da algoritmanın basit mantığını oluşturur ve hesaplama karmaşıklığını en aza indirir, ancak model bazı gerçek zorluklarda istenmeyen sonuçlara ulaşır (Juan ve ark., 2018).

Naif Bayes, X ve Z olmak üzere iki rastgele oluşumun olasılıklarını hesaplamak için kullanılan bir modeldir. Hipotez, maksimum olasılık ilkesini içerir, yani sınıflandırıcı, sınıfın üstün ve daha büyük olasılığa ait özelliğini tahmin ettiği anlamına gelir. X ve Z'nin rastgele olaylar olduğu varsayılarak Denklem 3.14'de gösterilmiştir.

$$P(X|Z) = \frac{P(X|Z)P(x)}{P(X)} \tag{3.14}$$

P(X), X'in bir bağımsızlık olasılığıdır, P(Z), Z'nin bir bağımsızlık olasılığıdır, P(X|Z) X olayının Z olayına bağlı olasılığını, P(x) önceki olasılığı, P(Z|X) Z olayının X olayına bağlı önceki olasılığını göstermektedir.

$$P(X_f|Z) = \frac{P(Z|X_f)P(X_f)}{P(x)} \tag{3.15}$$

Çoğunlukla veri kümesi, a x b vektörü gibi sembolize edilen çok sayıda örnekten oluşur; burada a örneklerin sayısı ve b etiketlerin sayısıdır. X'in, $X = \{x_1, x_2, x_3, x_4, \dots, x_n\}$'i sınıflandırmamız gereken veri kümesi öznitelikleri olduğunu varsayalım, naif bayes sınıflandırıcısı, X örneklerini X_f sınıfına göre hesaplar ve ardından en yüksek olasılık değerine sahip olan özelliği seçer.

Naif Bayes sınıflandırıcı tamamıyla varsayıma dayanır ve örnekler birbirinden bağımsızdır, X_f sınıfına sahip olan bir X niteliği ancak $P(X_f|\,Z) > p(X_n|\,D)$ koşulu doğruysa ikili sınıflandırma uygulanabilir. Örneklerin birleştirme olasılığı, her bir örneğin olasılık sonucuyla karşılaştırılabilir, Denklem 3.16'da birleştirme olasılığı denklemi gösterilmektedir.

$$P(X\,|Z) = P(x_1, x_2, x_3, x_4, \dots, x_n|Z) = \prod_{i=1}^{n} P(ai|Z) \tag{3.16}$$

Naif bayes sınıflandırıcısının önceki olasılığı, Denklem 3.17'de yer alan denklem kullanılarak hesaplanır.

$$P(X\,|Z) = P(X) = \prod_{i=1}^{n} P(ai|Z) \tag{3.17}$$

Maksimum olasılık, belirli bir D veri kümesinin olay veya Z sınıfı ile maksimum olasılığıdır. Denklem 3.18'de denklem gösterilmiştir.

$$P_{ML} = \text{argmax}P(D|Z) \tag{3.18}$$

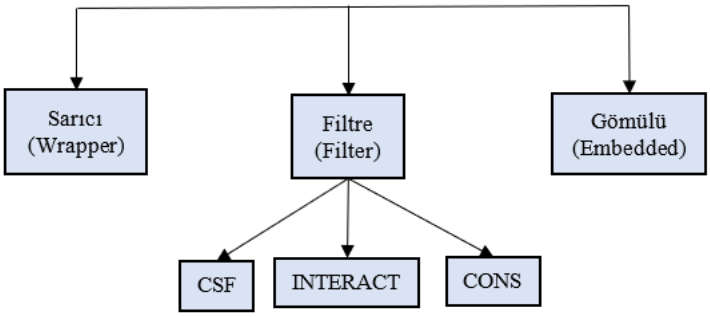
3.8. Özellik Seçme

Etkili bir saldırı tespit sistemi oluşturma amacına ulaşabilmek için, modelin verimliliğini artıran ve sistemin yüksek doğrulukta sonuçları geliştirmesini sağlayan özellik seçimi (FS) yöntemlerinin kullanılması gerekmektedir. Ayrıca özellik seçimi, hangi özelliğin diğerlerinden daha önemli olduğunu vurgular.

Özellik seçimi, veri kümesinden uygun özellikleri seçmektir, aynı zamanda, özellik seçiminin amacı, herhangi bir özelliği kaybetmeden orijinal veri kümesiyle aynı özelliğe sahip küçük alt kümeler halinde verileri en aza indirgemektir.

Özellik seçiminde kullanılan metotlar Filtreleme, sarmal (wrapper) ve gömülü (embedded) olmak üzere üç grupta toplanmaktadır. Filtre modelinde özellik seçimi, sınıflandırma algoritmalarından bağımsızdır; örneğin karşılıklı bilgi ve korelasyon katsayısı algoritmaları gibi. Sarmal yöntemi, diğer özellikler arasında öne çıkan özellikleri değerlendirmek için öğrenme (sınıflandırma) algoritmalarına bağlıdır. Gömülü (Embedded) yöntem hem filtreleme hem de sarmal yöntemlerin bir kombinasyonudur. Kompozisyon yöntemleriyle oluşturma işlemidir; Örneğin Düzenlileştirme yöntemleri ve LASSO. Arama alanı çok büyük olduğundan, özellik seçimi zorlaşmaktadır. Özellik seçimi belirlenirken hesaplama maliyeti yüksek olabilir (El-Khatib ve ark., 2010).

Sarmal(wrapper) yönteminde, seçilen özelliklerin önemine karar vermek için sınıflandırma algoritmaları kullanılır. Bu tür bir yöntemi çok sayıda örnekte çalıştırmanın maliyeti düşüktür, çünkü sınıflandırma algoritmalarını düzenli olarak yürütmek, seçilen özelliklerin önemini değerlendirmek için zorunludur. Gömülü yöntem hem filtreleme hem de sarmal yöntemlerin kusurunu giderir (Vajiheh ve Shahram, 2018).



Şekil.3.5. Özellik seçimi yöntemleri. (Vajiheh Hajisalem & Shahram, 2018).

3.8.1. Akıllı Su Damlaları Algoritması (IWD)

Akıllı su damlaları (Intelligent water drops (IWD)), suyun doğal sürüsünden türetilen yeni bir meta-sezgisel optimizasyon modelidir ve Shah Hosseini tarafından 2009 yılında önerilmiştir (Neha ve ark. 2013). Ayrıca, algoritmanın gözlemi nehir, göl veya okyanuslardan akan ılık sulara dayanmaktadır.

Bu algoritmanın asıl amacı, maliyeti düşürmektir. IWD'lerde, su damlaları yolu daha az toprakla izleme eğilimindedir, bu nedenle algoritma, diğer su damlalarını

çözümün ilgili yoluna çekmek için en iyi geliştirilmiş çözümlerin bileşenlerinden toprağı çıkarmaya çalışır. Algoritma, belirli bir sorunun ideal çözümünü elde etmek için Akıllı su damlalarının (IWD) sayısını kullanır. Herhangi bir problem için düğüm kümesi (S) ve kenar kümesi IWD akışı grafiğin kenarları boyunca ilerleyen bir yol veya yol oluşturmak için tasarlanır. Her akıllı su damlacığı, kendi çözümünü oluşturmak için grafiğin düğümlerini (S) kenarlarından (E) geçirir, bu işlem $T^{(IWD)}$ olarak belirtilen mutlak bir çözüme ulaşılmadıkça devam eder. Bu çözüm $T^{(IWD)}$) IWD'nin kenarlar boyunca uzandığı yolu ifade eder. Ayrıca, algoritma, bütün Akıllı su damlaları (IWD'ler) kararlılığa ulaştığında sona erer. T^{IB} Olarak adlandırılan her bir döngünün optimum çözümü aranır. Bir kalite fonksiyonu, diğer IWD'ler tarafından elde edilen çözümler arasında en uygun çözümü (T^{IB}) değerlendirir.

Alijla ve ark. (2013), bu algoritmayı özellik seçimi için kullanmışlardır. İlk aşama ilk değer atamasıdır. Bu aşamada statik ve dinamik parametreler ve grafik (problemin tamamı grafik olarak gösterilir) değerleri başlatılır. İkinci aşama, her bir su damlası için ayrı ayrı çözümü gerçekleyen çözüm aşamasıdır. Ayrıca bu aşama, Kenar seçimi ve değerlerin güncellenmesi olmak üzere temelde iki aşamadan oluşur. Kenar seçimi Denklem 3.19'da gösterilmektedir.

$$P_i^{IWD_k(J)} = \frac{f(soil(i,j))}{(\sum_{l=Vc} IWDk\ f(soil(i,l))} \tag{3.19}$$

$f(soil(i,j))$, i ve j arasındaki toprakların ters değerini hesaplarken. $soil(i,j)$ ise i ve j arasındaki toprak miktarının yerel yoludur.

Değerlerin güncellenmesi; bu aşamada suyun hızı ve su damlası her hareketi i noktasından j'ye değiştirir.

Üçüncü aşama, Yeniden Yapılanmadır, bu adımda, IWD tarafından kurulan optimum çözüm $T^{(IWD)}$ bu denklem kullanılarak hesaplanır. Dördüncü aşama sonlandırmadır, bu aşama birinci aşama ve ikinci aşama en iyi çözüme ulaşmadıkça tekrarlanır.

3.8.2. Gri Kurt Optimizasyonu (GWO)

Gri kurt optimizasyonu (Gray Wolf Optimization (GWP)) kurtların davranışlarından esinlenilen yeni bir meta sezgisel optimizasyon yöntemidir.

Algoritmanın davranış hiyerarşisi, doğada gri kurtların avlanma tekniğine benzemektedir. Gri kurt optimizasyonunda, alfa (α), beta (β), delta (ϱ) ve omega (ω) olmak üzere dört gri kurt kategorisi kullanılmaktadır.

Gri kurtlar 5-12 kişilik gruplara ayrılır, bu meta-sezgisel optimizasyon için alfa (α) diğerleri arasında anahtar parametredir, bu nedenle alfa optimum bir sonuç elde eder ve diğer parametreler önem sırasına göre sırasıyla ikinci, üçüncü ve dördüncü sırada yer almaktadır. Her zaman kurtların paket halinde avlanma prosedürü sırasında alfa (α), beta (β), delta (ϱ) avlanmaya (optimizasyon) rehberlik ederken, s (ω) üç kürtün izini sürmekten sorumludur. Bu sürecin ana bölümü, gri kurdun avı hemen çevrelemeye başlamasıdır. Çevreleyen davranış aşaması, Denklem 3.20'de ifade edilmiştir.

$$W(Iter+1) = P(Iter) - A.D_{(i)} \tag{3.20}$$

Iter, mevcut iterasyonu göstermektedir. P avın şu anki konumudur, A ise bir katsayıdır.

$$D_i = |C.P_{(t)} - W_{(t)}| \tag{3.21}$$

W gri kurtların şu anki pozisyonudur. C katsayı vektörüdür. A ve C, Denklem 3.22 ve 3.23'deki gibi hesaplanırlar.

$$A = 2b.r_1 - b \tag{3.22}$$

$$C = 2.r_2 \tag{3.23}$$

Burada, hem A hem de C katsayı vektörleridir, b 2'den 0'a kadar lineer olarak azalan bir bileşen değişkenidir, r_1 ve r_2 0 ile 1 aralığında değer alan rastgele vektörlerdir.

Ayrıca alfa, avlanma yönünü gösteren ve yönlendiren liderdir, beta, delta ve gama izdeyken alfaya avcılık için yardımcı olur. Bununla birlikte, arama haritasındaki avın konumu bilinmemektedir. Bu nedenle avın konumunu bulmak için matematiksel av eylemini taklit eden en uygun çözüm tahmin edilmektedir. Bu avlanma sürecinde beta ve delta daha çok ön plana çıkmaktadır.

3.8.3. Aslan Optimizasyonu (LOA)

Aslan Optimizasyonu Algoritması (Lion Optimization Algorithm (LOA)), 2015 yılında, aslanların doğal davranışları örneklenerek ortaya atılmıştır. Avlanma sürecinde aslanlar, avı tahrip etmek için özel bir saldırı eylem planı hazırlamaktadır. LOA, sürü tabanlı metasezgisel bir algoritmadır. Performansı, arama alanlarındaki popülasyon boyutunu kullanan diğer metasezsgisel algoritmalar gibidir. Başlangıç noktasında her aslanın kendi hızı ve konumu Denklem 3.24 ve 2.25'deki gibi hesaplanır.

$$L_{(i)} = (L_{(i,1),}L_{(i,2),}L_{(i,3),} \dots, L_{(i,n),})^2 \quad (3.24)$$

$$V_{(i)} = (V_{(i,1),}V_{(i,2),}V_{(i,3),} \dots, V_{(i,n),})^2 \quad (3.25)$$

$L_{(i)}$, aslanın başlangıç pozisyonu; $V_{(i)}$, aslanın başlangıç hızı; N ise popülasyon boyutudur. Her aslan hızını ve konumlarını güncellemek için Denklem 3.26, 3.27 ve 3.28 kullanılır.

$$freq_{(i)} = frq_{(min)} + (\ freq_{(max)} - freq_{(min)}\ ).freq_{(i)} \quad (3.26)$$

$$Vij^t = Vij^{t-1} + (\ Pij^{t-1} - P - best_{(j)}\ ).\alpha \quad (3.27)$$

$$Pij^t = Xij^{t-1} + Vij^t \quad (3.28)$$

Burada; α [0,1] arasında rasgele oluşturulmuş bir değerdir. Avlanma süreci boyunca aslanlar yerlerini değiştirebilirler, aynı zamanda aslan ve av arasındaki mesafe çok yaklaştığında yeni pozisyon için hızını arttırır.

$$Ci^{t+1} = \beta.Ci^t \quad (3.29)$$

$$ni^{t+1} = ni^0.[1 - ei^{-rt}] \quad (3.30)$$

β ve r statiktir, aralıkları 0> β >1 ve r >1 'dir. Her döngü için uygunluk değeri C ve n'ye göre artırılır. C 1'den başlar ve n ise $n_{(0)}$'da başlar.

Arama alanındaki yeni popülasyonları oluşturmak için üç temel koşula dikkat edilmelidir. Bunlar;

- Durum 1: Mevcut aslanlardan Yavru (Cubs) adında yeni bir çözüm oluşturulur.
- Durum 2: Yeni bir optimum çözüm (göçebe) elde etme sürecine Bölgesel Savunma denir
- En iyi çözümü kaybetmemek için Bölgesel Devralma olarak adlandırılır.

Özet olarak, Aslan Optimizasyonu Algoritmasının karar değişkenleri sayısı X, popülasyonun büyüklüğü (p), arama haritası (S), aslanın bölgedeki başlangıç yeri ve avının rasgele oluşturulması olmak üzere dört önemli parametresi vardır.

3.8.4. Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (Particle Swarm Optimization (PSO)), kuş ve balık sürülerinin davranışlarının modellenmesi ile oluşturulan, görevi en uygun çözümü elde etmek olan metasezgisel optimizasyon algoritmasıdır. PSO, Kennedy ve Eberhart (1995) tarafından önerilmiştir. Algoritma birçok farklı alanda uygulanmıştır.

PSO metodolojisi başlangıçta popülasyondaki parçacıklar rasgele pozisyonlara yerleştirilir. Her bir parçacık $\mathrm{Pbest}_{i,j}$ olarak adlandırılan kişisel en iyisi belirlenir. Her parçacığın konumu, parçacıklar arasında en iyi konumu seçmek için uygunluk fonkisyonu ile hesaplanır, daha sonra arama parçacıklarının yeni pozisyonu $X_{i,j}$, hızlarına (V_{ij}) göre güncellenir. Bu işlemler, $\mathrm{Gbest}_{i,j}$ anlamına gelen küresel en iyi pozisyon elde edilene kadar yürütülecektir. Parçacıkların konumunu ve hızını güncellemek için Denklem 3.31 ve 3.32 kullanılmaktadır.

$$V_{ij}(t+1) = wV_{ij}(t) + C_1R_{ij}(t)\left(Pbest_{ij} - X_{ij}(t)\right) + C_2R_2\left(\mathrm{Gbest}_{i,j} - X_{ij}(t)\right) \quad (3.31)$$

$$X_{ij}(t+1) = X_{ij}(t) + v_{ij}(t+1) \quad (3.32)$$

i parçacıklarının indeksini belirtir ve j ise boyut indeksi olarak tanımlanır. $V_{ij}(t+1)$ parçacığının bir sonraki iterasyondaki yeni hızı iken, $V_{ij}(t)$ şu anki hızını temsil eder. $X_{ij}(t+1)$ bir sonraki iterasyondaki yeni pozisyondur ve $X_{ij}(t)$ parçacığının mevcut konumunu göstermektedir. C_1 katsayılı (bilişsel) öğrenme faktörüdür ve C_2 sosyal öğrenmedir, genel olarak hem C_1 hem de C_2 ivmedir ve

değerleri 2'ye eşittir. R_1 ve R_2 , 0 ile 1 arasında rasgele oluşturulmuş birer değerdir. w, atalet ağırlığıdır.

İki yıl sonra Binary PSO (BPSO), Kennedy ve Eberhart tarafından önerilmiştir. Bu yeni teknik, bir özniteliğin seçildiği (1) veya göz ardı edildiği (0) olasılığını değiştirmek için ikili seçim uygulanmıştır, öznitelik seçim işlemi, ikili sigmod işlevi aracılığıyla hız değerlerine bağlıdır. Konum belirleme bu denklem kullanılarak gerçekleştirilir. (Kennedy ve Eberhart, 1997).

$$x_{ij}(t+1) = \begin{cases} 0 & if\ rnd > sig\,(v_{id}\,(t+1)) \\ 1 & otherwise \end{cases} \tag{3.33}$$

rnd [0, 1] arasında rasgele oluşturulmuş bir değerdir, sig ise Sigmoid fonksiyonu ifade eder. Denklem 3.34'de sigmoid fonksiyonu bulunmaktadır.

$$Sig(\,v_{id}\,(t+1)) = \frac{1}{(exp-(v_{id}\,(t+1)))} \tag{3.34}$$

3.9. Sinüs Kosinüs Algoritması (SCA) ve BSCA

Bazı özellik seçimi algoritmalarının, sınıflandırma doğruluğunun sonucunu etkileyen yüksek hesaplama maliyeti gibi zorlukları olmasına rağmen, bu sorunu düzeltmek için birçok araştırmacı, hesaplama maliyetlerini en aza indirmek için Sinüs Kosinüs optimizasyon algoritmaları (SCA) gibi evrimsel veya metasezgisel algoritmaları kullanmıştır. SCA, özellik seçimi dahil olmak üzere birçok alanda uygulanan popülasyon tabanlı metasezgisel bir algoritmadır. Bu çalışmada Binary SCA kullanmanın temel amacı, sınıflandırma performansını iyileştiren ve daha iyi sonuçlara ulaşan önemli bir özellik elde etmek için özellik seçimidir. Özelliğin seçilmesi durumunu ve seçilmemesi durumunu 1 ve 0 ile göstererek aday çözümler üretildiği için BSCA daha uygun bir temsil yaklaşımı olmuştur. Tez kapsamında kullanılan veri seti çok büyük ve birçok özelliğe sahip olduğu için Binary SCA tercih edilmiştir. Sınıflandırma işlemi için ise SVM kullanılmıştır. SCA'nın zaman karmaşıklığı $O(x * y * z_{sca})$ 'ya eşittir; burada x yineleme sayısını, y ajan sayısını, z ise SCA'nın aracıların konumunu hesaplaması için geçen süreyi göstermektedir (Mirjalili, 2016).

Sinüs kosinüs algoritması, Sinüs ve Kosinüs fonksiyonlarını temel alan popülasyon tabanlı bir optimizasyon algoritmasıdır. Algoritma, Mirjalili (2016) tarafından sunulmuştur. SCA, diğer metasezgisel algoritmalara benzemektedir. Algoritma şu şekilde işlemektedir. Ajanların rasgele konumlarda (çözümlerde) başlatılması, daha sonra en iyi konumunu elde etmek için arama haritasında rasgele aramaya başlaması ve her üretilen ajan için uygunluk fonksiyonunun hesaplanması. En iyi ajanın konumu, diğer ajanlar arasından seçilerek ve her tekrarda P (en iyi) olarak belirlenecektir. Son olarakta, en iyi ajanlar Denklem 3.35 kullanılarak güncellenecektir.

$$Pij^{(t+1)} = \begin{cases} Pij^t + n1 * sine(n2) * |n3\ Xj^t - Pij^t| & n4 < 0.5 \\ Pij^t + n1 * cos(n2) * |n3\ Xj^t - Pij^t| & n4 \geq 0.5 \end{cases} \quad (3.35)$$

$Pij^{(t+1)}$ Ajanın j boyutunda (t + 1) iterasyonuna ilişkin yeni güncellenmiş konumudur. Algoritmanın dört parametresi vardır. Bunlar; n1, n2, n3 ve n4, her birinin kendine özgü bir görevi vardır. Denklem 3.36 uygunluk fonksiyonunu nasıl hesapladığımızı göstermektedir.

$$Fitnes\ Function = M\ x\ Erro\ Rate + \frac{Selected\ features}{(Number\ of\ dataset\ features)} \quad (3.36)$$

Burada *M*, eşitleme faktörüdür.

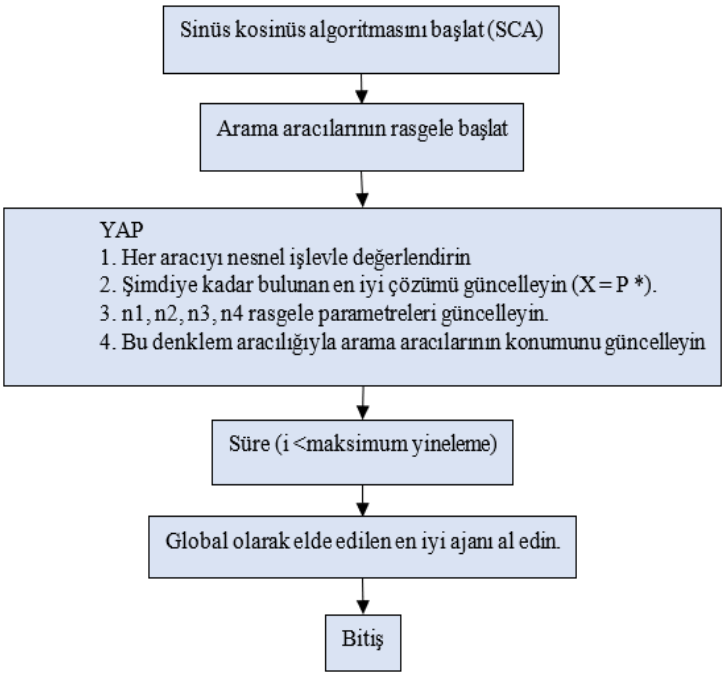
3.9.1. Sinüs kosinüs algoritmasının parametresi

n1 parametresi, aracının hedefe doğru veya hedeften uzaklaşarak gittiği yönü belirtmek için kullanılır. Kısaca n1, konum güncellendiğinde ajanın bir sonraki konumu nerede olacağına, yerel veya küresel aramada mı olacağına karar verir. n1 parametresi hareketin hedefe gidip gitmediğini nasıl belirler? 0 ile 1 arasında rasgele değerler kullanılarak, n1 <1 değeri, ajanın hedefe doğru gitmesi durumunda, n1> 1, ajanın hedeften dışarıya doğru hareket ettiği anlamına gelir. Bu, algoritma aşamalarının keşfi ve kullanımı arasındaki dengeyi kontrol etme işleminde n1'in sorumlu olmasının temel nedenidir. Dengeleme aşamasında n1'in iterasyon döngüsü sabit değerden 0'a doğrusal olarak düşer (Hafez ve ark., 2016).

$$n1 = c - t\frac{c}{Tmax} \quad (3.37)$$

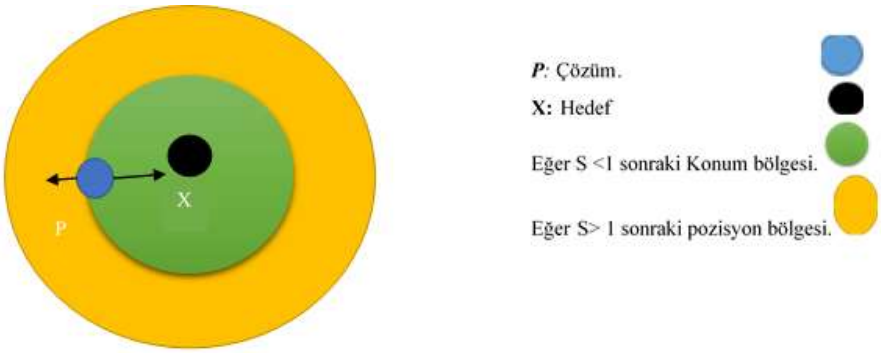
C sabit değerdir, mevcut iterasyon t olarak gösterilir ve T_{max} maksimum döngüdür.

n2 parametresi, ajanın hedefe ne kadar yakın veya uzak olduğunu belirtir. n3 parametresi, mesafenin tanımlanmasında hedefin etkisini stokastik olarak vurgulamak (r3> 1) veya önemsizleştirmek (r3 <1) için hedef için rastgele bir ağırlık getirir. Dördüncü parametre n4, sinüs ve kosinüsü eşit bir şekilde anahtarlamaktan sorumludur.



Şekil. 3.6 SCA'nın akış dyagramı.

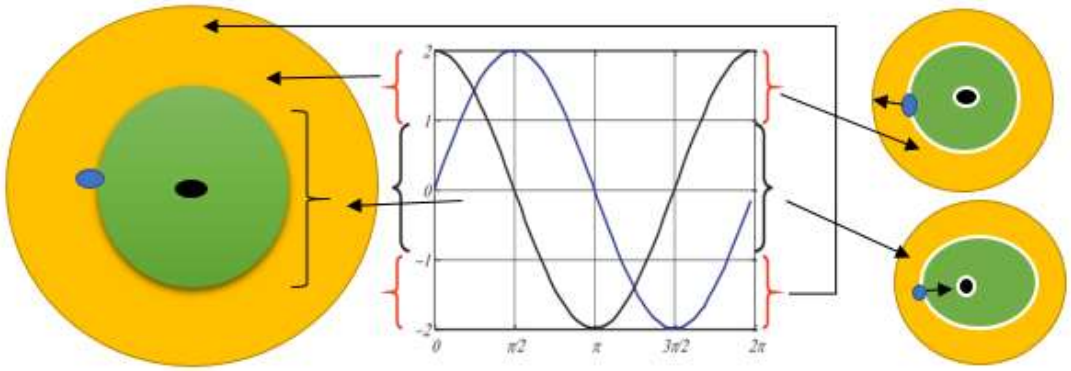
3.9.2. Sinüs kosinüs algoritmasının iki boyutlu arama mekanizması



Şekil 3.7. Sinüs kosinüsünü denklemde gösterildiği gibi bir sonraki konumda güncellemek. (3.35) ,(Seyedali Mirjalili, 2016).

Yukarıdaki özellik, sinüs ve kosinüs algoritmasının sunulan denklemlerinin, biri aracı, diğeri ise hedef olan iki çözüm arasındaki arama alanına nasıl uygulandığını göstermektedir. Algoritmanın denklemi yalnızca Şekil 3.7'deki gösterimde olduğu gibi sahip olunan iki boyutu uygularken değil aynı zamanda çoklu boyutlarda da uygulanabilir. Sinüs ve kosinüs döngüsel modele göre, bir çözüm birbiri ardına yeniden konumlandırılabilir, böylece bu, iki çözüm arasındaki boşluğu sömürüye dönüştürür. Böylece bu, iki çözelti arasındaki boşluğun sömürüye sahip olmasını sağlar. Çözüm, arama alanını keşfetmek için karşılık gelen varış noktaları arasındaki boşluğun dışına çıkabilir, bunu yapmak için sinüs ve kosinüs fonksiyonlarının sınırını değiştirmemiz gerekir.

Şekil 3.8'de sinüs ve kosinüs algoritma ajanlarının çözümü iki boyutlu [-2, 2] uzayda nasıl araştırdığı gösterilmektedir. s-1 ile 1 arasında çözüm hedefe yaklaşır (kefaret aşaması). -1 ila -2 ve 1 ila 2 arasında çözüm, küresel arama alanına (keşif aşaması) gider (Mirjalili, 2016).



Şekil. 3.8. Sine Cosine [-2,2] aralığısndaki ajanların hedefe doğru veya onları aşağılamasını sağlar. (Seyedali Mirjaslili, 2016).

3.9.3. İkili sinüs kosinüs algoritması (BSCA)

Bu çalışmadaki temel, en iyi sonuca ulaşmak için özellikleri en az indirgemek ve sınıflandırma performansını arttırmaktadır. Sorunu çözmek için, Sinüs-Kosinüs algoritmasının gerçek değer sayılarını ikili değerlere dönüştürmek (Binary Sine Cosine algorithm, BSCA) gerekmektedir. İkili dönüştürülmüş varyantlarda hem arama alanı hem de ajanların konumu ikili dize şeklindedir. İkili Tanh dönüşüm fonksiyonu ve benzeri birçok fonksiyon dönüşümü olmasına rağmen (Srikanth ve ark., 2017), bu

çalışmada Sigmoid Dönüşüm fonksiyonu kullanılmıştır. Bu fonksiyon da ikili değerlerin seçilmesiyle ilgili dönüşüm fonksiyonlarından biridir. Denklem 3.38'de fonksiyon yer almaktadır.

$$P_{ij}(t+1) = \begin{cases} 1 & S(\Delta(t+1)) > O \\ 0 & Otherwise \end{cases} \quad (3.38)$$

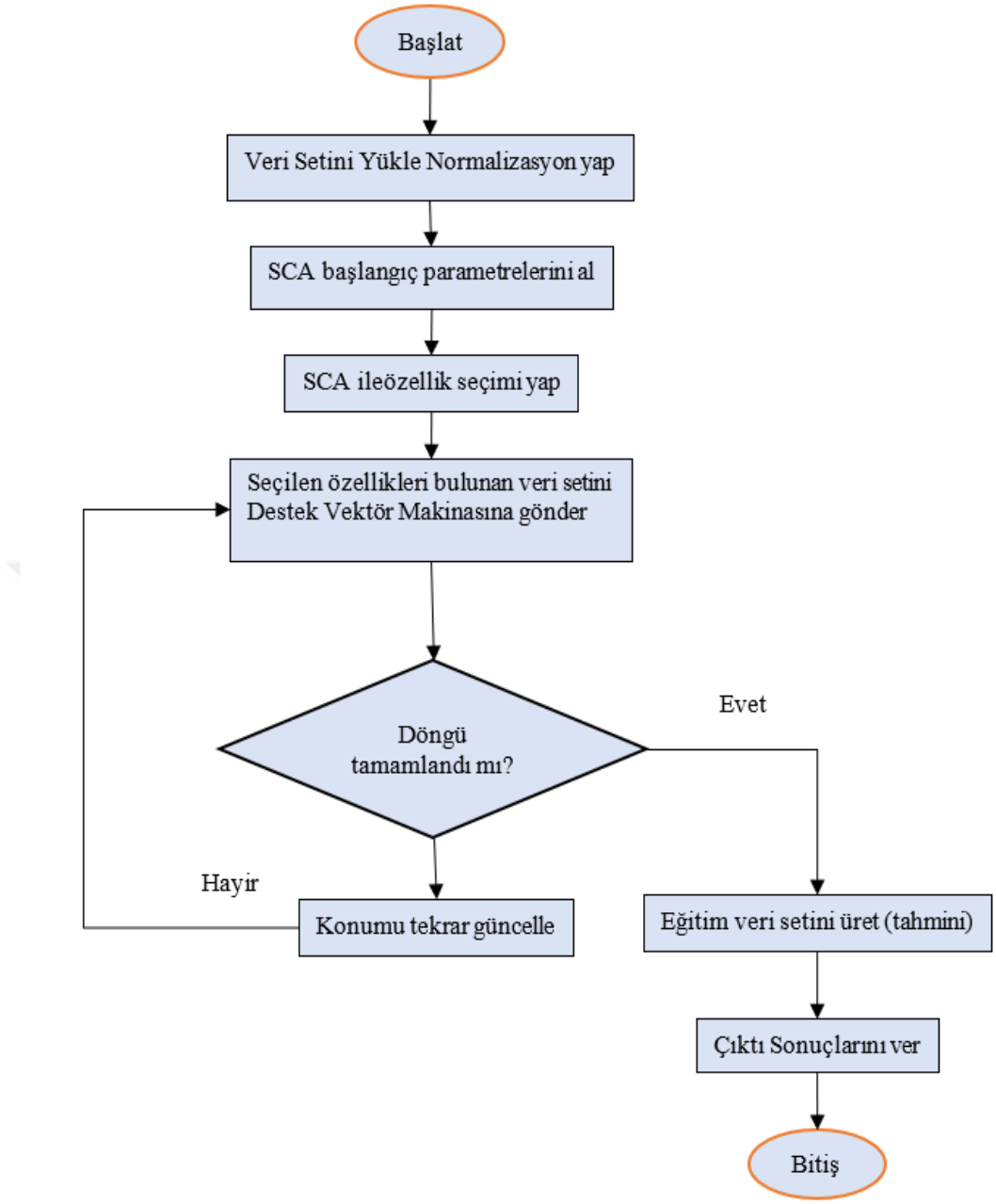
$S(\Delta(t+1))$ Güncellenmiş konumu gösterir ve O, 0 ile 1 arasında dağıtılmış bir sayıyı temsil etmektedir.

$$S(\Delta(t+1)) = Sig(\Delta(t+1)) = \frac{1}{(1+exp-(\Delta(t+1)))} \quad (3.39)$$

3.10. Destek Vektör Makinesi ile ikili sinüs kosinüs algoritması (BSCA-SVM)

Tez çalışmasında, ikili SCA (Binary Sine Cosine algoritması (BSCA)) seçilmesinin nedeni, iyi bir arama özelliğine sahip olmasıdır. Buna ek olarak algoritmanın, diğer özelliklerin alt kategorisini hesaplayan ve daha sonra Destek Vektör Makinesine geçen uygunluk fonksiyonuna sahip olmasıdır. En iyi doğrulukla SVM'nin sınıflandırma yapması için, en uygun özellik seçimi algoritması olarak BSCA tercih edilmiştir.

Şekil 3.9'da BSCA-SVM'nin akış şeması gösterilmektedir. Algoritma giriş veri kümesini alır, popülasyon boyutunu ve parametreleri ilk değer atamaları yapılır. Daha sonra popülasyon normalleştirme denklemini kullanarak ikili biçime dönüştürülür, Uygunluk fonksiyonu ile algoritma sonunda optimum seçimi yapar ve daha sonra özellikleri hesaplamak için sınıflandırma algoritmasına geçecektir. Her iki algoritma (BSCA ve SVM), programdan çıkış koşulu sağlanana kadar parametreleri ve aracıların yeni konumunu düzenli olarak güncellemeye devam edecektir. Algoritma sonlandığında, program tahmin edilen çıktıyı ve test verilerini karşılaştırır, daha sonra sonucu yazdırır.



Şekil.3.9 BSCA-SVM temel akış diyagramı.

3.11. Veri Kümeleri

Sistemin performansından emin olmak ve test etmek için uygun bir veri setinin seçilmesi gerekmektedir. Bununla birlikte, internetten veya ağ üzerinden yeni bir veri seti oluşturulmak hem oldukça maliyetli hem de verimsiz olacaktır. Buna ek olarak, çeşitli kaynaklardan topladığımız verilerde istenilen tipik saldırı gibi aradığımız tüm bilgiler mevcut olmayabilir. Bu nedenle sistemimizi test etmek için uygun ve literatürde

kullanılan bazı veri setleri bulunmaktadır. Literatürde uygulanan en popüler ve verimli veri setleri NSL-KDD ve UNSW-NB15'dir.

3.11.1. NSL-KDD veri seti

NSL-KDD, Stolfo ve lee tarafından önerilen KDD CUP99'un orijinal standardına özgüdür (Acharya ve Singh ,2017). Ayrıca, KDD CUP99, Birleşik Devletler Hava Kuvvetleri ağında uygulanmış anomali tespiti için en çok tercih edilen ve kullanılan veri setlerinden biridir. Ancak 2009 yılında Tavallaee ve arkadaşları bu veriler üzerinde çalıştıklarında, sistemin değerlendirme performansını etkileyen ve yavaşlatan iki önemli sorunla karşılaşmışlardır. Bu zorlukları gidermek için NSL-KDD sunulmuştur. NSL-KDD, orijinal olarak KDD-CUP99'dan alındığı için aynı KDD-CUP99 yapısına sahiptir, her iki veri kümesi de aynı saldırı türlerine sahiptir ve özellikler de benzerdir, ancak NSL-KDD yeni ve özlü olanıdır.

Moustafa ve Jill (2015)'e göre NSL-KDD'yi geliştirmenin arkasında üç amaç vardır. Birincisi, KDD veri kümesinin hem eğitimini hem de testini yineleyen satırları ortadan kaldırmaktır. İkincisi, sınıflandırma performansını artırmak ve başarılı sonuçlar elde etmek için KDD veri kümesinin farklı bölümlerinden çeşitli kayıtlar almaktır. Üçüncü hedef ise, dengesizlik komplikasyonunu hem eğitim hem de test setinden çıkararak hata oranını en aza indirgemek, bu Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1. NSL-KDD'nin katkı dağıtımı (Moustafa ve Jill, 2015).

NSL-KDD veri kümesi		
Sınıflar	Eğitim verileri	Test verileri
Normal	67,343	9,710
Denial of service	45,927	7,458
Remote to Local	995	2,887
Probe	11,656	2,422
User to root	52	67
Total Instances	**125,973**	**22,544**

Çizelge 3.2’de, her bir özelliğin adı ile birlikte NSL-KDD veri setini göstermektedir. Özellik tanımlarıyla ve sürekli mi yoksa sembolik mi olduklarını gösterir.

Çizgelge 3.2. NSL-KDD'nin özellik dağılımı (Moustafa ve Jill, 2015)

Veri kümesi alan isimleri	Niteliklerin Açıklaması (ingilizce)	Niteliklerin Açıklaması	Nitelik (Inglizce)	Nitelik
S	Connection time (sec.)	Bağlantı süresi (sn.)	Continuous	Sürekli
Prtocol_type.	Protocol type, e.g., TCP, UDP, and so on.	Protokol türü, ör. TCP, UDP vb.	Symbolic	Simgesel
Service	Target network service, e.g., HTTP, telnet, etc.	Hedef ağ hizmeti, ör. HTTP, telnet vb.	Symbolic	Simgesel
Flag	The status of connection: normal or error.	Bağlantının durumu: normal veya hata	Symbolic	Simgesel
Src_bytes	Amount of data transmitted from the source to the destination (byte).	Kaynaktan hedefe aktarılan veri miktarı (bayt).	Continuous	Sürekli
Dst_bytes	The amount of data transmitted to the source (byte).	Kaynaktan hedefe aktarılan veri miktarı (bayt).	Continuous	Sürekli
Land	If the source and destination of the connection are the same server/port 1, other 0.	Bağlantının kaynağı ve hedefi aynı sunucu / port 1 ise, diğer 0.	Symbolic	Simgesel
Wrong_fragment	Number of defective parts.	Arızalı parça sayısı.	Continuous	Sürekli
Urgent	Emergency package number.	Acil durum paket numarası.	Continuous	Sürekli
Hot	“Hot” Number of indicators.	"Sıcak" Gösterge sayısı.	Continuous	Sürekli
Num_ failed_logins	Number of failed logins.	Başarısız oturum açma sayısı	Continuous	Sürekli
Logged_in	Successful entry 1;	Başarılı giriş 1; Diğer 0	Symbolic	Simgesel.

	Other 0			
Num_compromised	dangerous "situation Number	dangerous "situation Number	Continuous	Sürekli
Root_shell	If the root shell is acquired 1; Other 0	If the root shell is acquired 1; Other 0	Discrete	Ayrık
Su_attempted	If the “su root " command is run 1; Other 0	“Su root " komutu çalıştırılırsa 1; Diğer 0	Discrete	Ayrık
Num_root	root "access number	kök "erişim numarası	Continuous	Sürekli
Num_file_creations	Number of files created	Oluşturulan dosya	Continuous	Sürekli
Num_shells	The number of open shell	Açık kabuk sayısı	Continuous	Sürekli
Num_access_files	Number of transactions in access control files	Erişim kontrol dosyalarındaki işlem sayısı	Continuous	Sürekli
Num_outbound_cmds	Number of outgoing commands in the FTP session	FTP oturumundaki giden komutların sayısı	Continuous	Sürekli
Is_hot_login	If first entry of the list is 1, other is 0	Listenin ilk girişi 1 ise diğeri 0'dır	Symbolic	Simgesel
Is_guest_login	"Guest" entry is 1; Other 0	"Misafir" girişi 1; Diğer 0	Symbolic	Simgesel
Num_compromised	dangerous "situation Number	dangerous "situation Number	Symbolic	Simgesel
Count	Number of connections made to the same server in t sec	T sn içinde aynı sunucuya yapılan bağlantı sayısı	Continuous	Sürekli
Srv_count	Number of connections made within the same service t seconds	Aynı hizmet içinde yapılan bağlantı sayısı t saniye	Continuous	Sürekli
Serror_rate	Percentage of connections received with an “SYN”	"SYN" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli

	error			
Srv_serror_rate	Percentage of connections received with an “SYN” error	"SYN" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli
Rerror_rate	Percentage of connections received with "REJ" error	"REJ" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli
Srv_rerror_rate	Percentage of connections received with "REJ" error	"REJ" hatasıyla alınan bağlantıların yüzdesi	Srv_diff_host_rate	Sürekli
Same_srv_rate	Percentage of connections made to the same service	Aynı hizmete yapılan bağlantıların yüzdesi	Continuous	Sürekli
Diff_srv_rate	Percentage of connections made to different services	Farklı hizmetlere yapılan bağlantıların yüzdesi	Continuous	Sürekli
Srv_diff_host_rate	Percentage of connections made to different servers	Farklı sunuculara yapılan bağlantıların yüzdesi	Continuous	Sürekli
Dst_host_count	Number of connections made in n connections to the same server	Aynı sunucuya yapılan n bağlantıda yapılan bağlantı sayısı	Continuous	Sürekli
Dst_host_srv_count	Number of connections made in the same service n connections	Aynı hizmette yapılan bağlantı sayısı n bağlantı	Continuous	Sürekli
Dst_host_same_srv_rate	Percentage of connections made to the same service	Aynı hizmete yapılan bağlantıların yüzdesi	Continuous	Sürekli
Dst_host_diff_srv_rate	Percentage of connections made to	Farklı hizmetlere yapılan bağlantıların	Continuous	Sürekli

	different services	yüzdesi		
Dst_host_same_src_port_rate	Percentage of connections from the same source port	Aynı kaynak bağlantı noktasından bağlantı yüzdesi	Continuous	Sürekli
Dst_host_srv_diff_host_rate	Percentage of connections made to different servers	"SYN" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli
Dst_host_serror_rate	Percentage of connections received with an “SYN ’” error	"SYN" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli
Dst_host_srv_serror_rate	Percentage of connections received with an “SYN” error	"SYN" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli
Dst_host_reror_rate	Percentage of connections received with "REJ" error	"REJ" hatasıyla alınan bağlantıların yüzdesi	Continuous	Sürekli

3.11.2. UNSW-NB15 veri seti

UNSW-NB15, NSL-KDD ve KDD99 gibi diğer veri kümelerinden daha fazla avantaja sahip olan, modern normal ve saldırı türlerini içeren saldırı tespit veri setinin yeni bir versiyonudur. Veri seti, normal ve saldırı türlerinin modern tasarımını yapmak için IXIA aracı kullanılarak ACCS (Avustralya Siber Güvenlik Merkezi) tarafından toplanmıştır (Mustafa ve Jill, 2015). Ayrıca, veri kümesinde 49 özellik ve 9 çeşit modern saldırı vardır. UNSW-NB15 kayıtları sembolik, tamsayı ve ikili satırlar halindedir. Ek olarak on iki algoritma, bağlantı paketlerinin akışlarını derinlemesine analiz etmek için C # dili kullanılarak geliştirilmiştir. Araçlar Bro-IDS ve Argus araçlarıdır. Araştırmacılar, IXIA aracını kullanarak normal ve anormal olan 9 tür özellik oluşturmuşlardır. Çeşitli özellik adları sınıflardan oluşur; Normal (Nor), hizmet reddi (Dos), analiz (Anl), Solucanlar (Wor), arka kapı (Bdr), shellcode (Shc), Generic (Gn),

Exploits (Exp), Fuzzers (Fuz) ve Keşif (Rec). Bu özelliklerin dağılımları Çizelge 3.3'te sunulmuştur.

Çizelge 3.3. UNSW-NB15'in özellik dağılımı (Moustafa, 2015).

UNSW-NB15 veri kümesi		
Sınıflar	Eğitim veri seti	Veri setini test etme
Nor	56,000	37,000
Dos	12,264	4089
Anl	2,000	677
Wor	130	44
Bdr	1,746	583
Shc	1,133	378
Gn	40,000	18,871
Exp	33,393	11,132
Fuz	18,184	6,062
Rec	10,491	3,496
Total Instances	**175,341**	**82,332**

Veri setinin tamamı iki bölümden oluşan eğitim ve test bölümlerine ayrılmıştır. Veriler hem test hem de eğitim seti olarak ayrı ayrı hazırlanmıştır. Çizelge 3.3'de belirtildiği gibi yaklaşık 93.000 (Normal), 16353 (denial of service), 2677 (analysis), 174 (Worms), 2599 (backdoor), 1511 (shellcode), 58871 (Generic), 44525 (Exploits), 24246 (Fuzzers) ve 13987 (Reconnaissance) veri bulunmaktadır. Çizelge 3.4'de UNSW-NB15 veri setinin açıklaması gösterilmektedir.

Çizelge 3.4. UNSW-NB15'in dağılımı (Moustafa, 2015).

Veri kümesi alan isimleri	Niteliklerin Açıklaması (ingilizce)	Niteliklerin Açıklaması	Nitelik (Inglizce)	Nitelik
Srcip	Source IP address	Kaynak IP Adresi	Nominal	Nominal
Sport	Source port number	Kaynak bağlantı noktası numarası	Integer	Tamsayı
Dstip	Destination IP address	Hedef IP adresi	Nominal	Nominal
Dsport	Destination port number	Hedef bağlantı noktası numarası	Integer	Tamsayı
Proto	Transaction protocol	İşlem protokolü	Nominal	Nominal
State	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS,	Durumu ve bağımlı protokolünü belirtir, ör. ACC, CLO,	Nominal	Nominal

	PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)	CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN ve (-) (kullanılmıyorsa durum)		
Dur	Record total duration	Toplam süreyi kaydedin	Float	Kayan
Sbytes	Source to destination transaction bytes	Kaynaktan hedefe işlem baytları	Integer	Tamsayı
Dbytes	Destination to source transaction bytes	İşlem baytlarının kaynağı için hedef	Integer	Tamsayı
Sttl	Source to destination time to live value	Değeri yaşamak için kaynaktan hedefe zaman	Integer	Tamsayı
Sttl	Source to destination time to live value	Değeri yaşamak için kaynaktan hedefe zaman	Integer	Tamsayı.
Dttl	Destination to source time to live value	Yaşam değeri için zaman kaynağı olan hedef	Integer	Tamsayı
Sloss	Source packets retransmitted or dropped	Kaynak paketler yeniden iletildi veya bırakıldı	Integer	Tamsayı
Dloss	Destination packets retransmitted or dropped	Hedef paketler yeniden iletildi veya bırakıldı	Integer	Tamsayı
Service	http, ftp, smtp, ssh, dns, ftp-data, irc and (-) if not much used service	http, ftp, smtp, ssh, dns, ftp-data, irc ve (-) çok kullanılmayan servis	Nominal	Nominal
Sload	Source bits per second	Saniyedeki kaynak bit sayısı	Float	Kayan
Dload	Destination bits per second	Saniyedeki hedef bit sayısı	Float	Kayan
Spkts	Source to destination packet count	Kaynaktan hedefe paket sayısı	Integer	Tamsayı
Dpkts	Destination to source packet count	Kaynak paket sayısının hedefi	Integer	Tamsayı
Swin	Source TCP window advertisement value	Kaynak TCP penceresi reklam değeri	Integer	Tamsayı
Dwin	Destination TCP window advertisement value	Hedef TCP penceresi reklam değeri	Integer	Tamsayı
Stcpb	Source TCP base sequence number	Kaynak TCP temel sıra numarası	Integer	Tamsayı
Dtcpb	Destination TCP base	Hedef TCP	Integer	Tamsayı

	sequence number	temel sıra numarası		
Smeansz	Mean of the row packet size transmitted by the src	Src tarafından iletilen satır paket boyutunun ortalaması	Integer	Tamsayı
Dmeansz	Mean of the row packet size transmitted by the dst	Dst tarafından iletilen satır paket boyutunun ortalaması	Integer	Tamsayı
Trans_depth	Represents the pipelined depth into the connection of http request/response transaction	Http talebi / yanıt işleminin bağlantısına ilişkin ardışık düzen derinliğini temsil eder	Integer	Tamsayı
res_bdy_len	Actual uncompressed content size of the data transferred from the server’s http service	Sunucunun http hizmetinden aktarılan verilerin gerçek sıkıştırılmamış içerik boyutu.	Integer	Tamsayı
Sjit	Source jitter (mSec)	Kaynak titreşimi (mSec)	Float	Kayan
Djit	Destination jitter (mSec)	Hedef seğirmesi (mSn)	Float	Kayan
Stime	record start time	kayıt başlangıç zamanı	Timestamp	Zamandamgası
Ltime	record last time	son kez kaydet	Timestamp	Zamandamgası
Sintpkt	Source interpacket arrival time (mSec)	Kaynak arası paket varış zamanı (mSec)	Float	Kayan
Dintpkt	Destination interpacket arrival time (mSec)	Hedef interpacket varış zamanı (mSn)	Float	Kayan
Tcprtt	TCP connection setup round-trip time, the sum of ’synack’ and ’ackdat’.	TCP bağlantısı kurulumu gidiş dönüş süresi, "eşzamanlama" ve "ackdat" değerlerinin toplamı.	Float	Kayan
Synack	TCP connection setup time, the time between the SYN and the SYN_ACK packets.	TCP bağlantısı kurulum süresi, SYN ve SYN_ACK paketleri arasındaki süre.	Float	Kayan
Ackdat	TCP connection setup time, the time between the SYN_ACK and the ACK packets.	TCP bağlantısı kurulum süresi, SYN_ACK ve ACK paketleri arasındaki süre.	Float	Kayan
Is_sm_ips_ports	If source (1) and	Kaynak (1) ve	Binary	İkili

	destination (3) IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0	hedef (3) IP adresleri eşitse ve bağlantı noktası numaraları (2) (4) eşitse, bu değişken 1 değerini alır, yoksa 0		
Ct_state_ttl	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).	Kaynak / hedef süresi için belirli değerler aralığına göre her durum (6) için numara (10) (11).	Integer	Tamsayı
Ct_flw_http_mthd	No. of flows that has methods such as Get and Post in http service.	Http hizmetinde Al ve Gönder gibi yöntemlere sahip akış sayısı.	Integer	Tamsayı
Is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.	Ftp oturumuna kullanıcı ve parola ile erişiliyorsa, 1 başka 0.	Binary	İkili

4. DENEYSEL ÇALIŞMA

Hızlı gelişen teknoloji ve kolay internet erişimi nedeniyle insanlar hem teknolojiye hem de internete büyük ölçüde bağımlı hale gelmiştir. Bu alışkanlığın tezahürü olarak, hassas bilgileri çalmak, bunları değiştirmek veya hatta verileri yönlendirmek için sistemlere kötü amaçla erişmeye çalışan saldırganlarla karşı karşıya kalmaktadır. Tez çalışmasında, güvenlik açıklarının bir kısmının üstesinden gelmek için, yeni bir saldırı tespit sistemi geliştirilmeye çalışılmıştır.

Bu çalışmada İkili Sinüs Kosinüs Algoritması (BSCA) ve Destek vektör makinesine (SVM) dayalı bir hibrit saldırı tespit sistemi gerçekleştirilmiştir. BSCA, özellik seçimi için ve SVM sınıflandırma için kullanılmıştır. Yeni modelin sonucu, doğruluk (accuracy) ölçütü dikkate alınarak aynı veri kümesini kullanan klasik SVM (RBF çekirdek fonksiyonlu ve Polinomsal çekirdek versiyonlu olarak), PSO-SVM, RF,k-NN, DT ve NB ile karşılaştırılmıştır.

Çalışmada UNSW-NB15 ve NSL-KDD veri kümeleri kullanılmış, veri kümelerinden rasgele ve eşit miktarda veri seçilmiştir. 2000, 1600, 800, 600, 400, 200 ve 100 veri örnekleri gibi farklı veri boyutu örneklerinde test edilmiştir. Önerilen yöntem veriler küçük olduğunda daha iyi sonuçlar vermektedir, bu nedenle 100 veri örneği alma da tercih edilmiştir. Küçük veri boyutu örneklerini almanın bir başka avantajı da RBF çekirdek işlevidir. Ayrıca büyük veri kümeleri üzerindeki bir bilgi işleme faaliyeti çok fazla zaman almaktadır. Bu nedenle de her iki veri kümesinden 100 veri örneği alınmıştır. Önerilen saldırıyı eğitmek ve test etmek için çapraz doğrulama bölme (Cross Validation partition(cvpartition)) tekniği kullanılmaktadır, veri setini eğitim için % 90 ve test için 10% 'a olarak ayarlandı.

Veri kayıtları kısaltma veya sembolik değerler içerdiğinden veri önişleme işlemleri kullanılmıştır. Veri dönüştürme ve normalleştirme işlemleri uygulanmıştır. Öncelikle sembolik değerler sayısal hale dönüştürülmüştür. Örneğin; Protokol niteliği tcp, udp ve icmp yi içeren bir sembolik özelliktir. Dolayısıyla, protokol niteliğinde sembolik değer sayısal değere; tcp = 1, udp = 2 ve icmp = 3 şeklinde dönüştürülmüştür. Benzer şekilde UNSW-NB15 veri kümesi için de aynı işlemler yapılmıştır. Tüm veri örneklerine, Denklem 4.1 ve 4.2'de gösterildiği gibi [0, 1] veya [-1, 1] aralık değerlerine dönüştüren Min ve Maks normalleştirme uygulanmıştır.

$$V = \frac{V - Min_A}{Max_A - Min_A}(newMax - newMin) + newMin \quad (4.1)$$

$$V = \left(\frac{V - Min_A}{Max_A - Min_A}\right) * (2 - 1) \quad (4.2)$$

Burada V değişkeni v'nin normalize edilmiş değeridir. Max_A ve Min_A değerleri ise gözlenen özelliklerin sırasıyla maksimum ve minimum değerleridir.

4.1. Performans Ölçüleri ve Tanımları

Metrikler, herhangi bir yeni sistemin performansını değerlendirmek için test edilmesinde önemli bir rol oynar. Temel olarak sınıflandırma sonucu aşağıdaki gibi tarif edilen dört sınıfa ayrılır;

- Gerçek pozitif: Bir saldırı olduğunda anormal olarak tespit edilen gerçek saldırı sayısını belirtir.
- Gerçek negatif: normal olarak sınıflandırılan normal miktarını tanımlar.
- Yanlış pozitif: normal olarak sınıflandırılan saldırı miktarını gösterir.
- Yanlış negatif: saldırı olarak tespit edilen normal miktarını gösterir.

Önerilen sistemin performansını değerlendirmek için bazı matris değerlendirmelerini dikkate almak gerekmektedir. Bunlar bu çalışmada kullanılan ölçütlerdir: Doğruluk (ACC), Duyarlılık (SEN), Özgüllük (SPE), Hassasiyet (PRE), F-skor (FM), Yanlış Pozitif Oran (FPR) ve Hata oranı (ERR) (Accuracy (ACC), Sensitivity (SEN), Specificity (SPE), Precision (PRE), F- measurement (FM), False Positive Rate (FPR), ve Error rate (ERR)). Bu metrikler Denklem 4.3, 4.4, 4.5, 4.6, 4.7, 4.8 ve 4.9'da gösterilmiştir.

$$\text{Doğruluk (ACC)} = \frac{tp + tn}{tp + tn + fp + fn} \quad (4.3)$$

Doğru (Accuracy) doğru sınıflandırılan veri sayısı, tüm veri sayısına bölünür. Modelin performansını hesaplamak için test verileri kullanılmıştır. En iyi doğruluk 1.0 iken en kötüsü 0.0'dır.

$$\text{Duyarlılık (SEN)} = \frac{tp}{tp + fn} \quad (4.4)$$

Hem duyarlılık hem de geri çağırma formül denklemi benzerdir. Bu, doğru pozitif türde sınıflandırılan örnek sayısının tüm pozitif örneklere bölünmesiyle elde edilir.

$$\text{Özgüllük (SPE)} = \frac{tn}{tn+fp} \tag{4.5}$$

Specificity (Özgüllük), sistemin negatif sınıflandırılan doğru veri sayısının gerçek değerinin negatif olduğu veri sayısına bölünmesi anlamına gelir.

$$\text{Hassasiyet (PRE)} = \frac{tp}{tp+fp} \tag{4.6}$$

Precision (Kesinlik) değerlendirme matrisi performans unsurlarından biridir, pozitif veri setine göre pozitif olarak sınıflandırılmış veri örneklerindendir.

$$\text{F} - \text{skor (FM)} = 2 * \frac{\text{Precision} * \text{Sensitivity}}{\text{Precision} + \text{Sensitivity}} \tag{4.7}$$

İkili sınıflandırmada, istatistiksel değerlendirmede F-ölçüm, hassasiyet ve geri çağırmanın harmonik bir ortalaması olarak tanımlanır. Başka bir F-ölçüm bölgesi şu şekilde tanımlanabilir: F-ölçüm, hassasiyet çarpı kesinlik hassasiyet artı kesinliğe bölünmesiyle elde edilir, hem üst kısım hem de alt kısım 2.0 ile çarpılır.

$$\text{Yanlış Pozitif Oran (FPR)} = \frac{fp}{tn+fp} \tag{4.8}$$

Yanlış pozitif Oran, tüm negatif veri kümeleri üzerindeki yanlış pozitif tahminlerin sayısıdır. Bu çok önemlidir, bu nedenle en iyi hata 1.0 iken en iyi yanlış pozitif oranın 0.0 olduğunu unutmamak gerekmektedir.

$$\text{Hata oranı (ERR)} = \frac{fp}{tn+fp} \tag{4.9}$$

Hata oranı (ERR), veri kümesinin tamamındaki tüm yanlış tahminlerin bağlanması olarak hesaplanan en iyi matris değerlendirmelerinden biridir. En iyi hata oranı 0.0, en kötü hata oranı ise 1.0'dır.

BSCA'nın SVM modeli ile sınıflandırma performansını öğrenmek için, yukarıda açıklanan doğruluk, özgüllük, duyarlılık, hassasiyet, F-Skor, yanlış pozitif oranı ve hata oranı gibi ortak performans ölçütleri hesaplanmıştır. Karışıklık matrisi, örnek verilerdeki gerçek sınıf etiketleri ile tahmin edilen sınıfların sayısından oluşur. İki sınıflı bir karışıklık matrisi örneği Çizelge 4.1'da gösterilmiştir.

Çizelge 4.1. İki sınıfa ait karışıklık matrisinin tanımı.

	TAHMİN		TOPLAM
	Normal	Saldırı	
Normal sınıf	TP	FN	Gerçek pozitif tutarı.
Saldırı sınıfı	FP	TN	Gerçek negatif miktarı
Toplam	Olumlu tahmin numarası	Olumsuz Tahmin Numaraları	Tüm örnek sayısı

Çizelge 4.2'de tez kapsamında kullanılan platformun (Bilgisayar & Matlab versiyonu olmak üzere) özellikleri gösterilmektedir.

Çizelge 4.2. Tez kapsamındaki çalışmaların yürütüldüğü ortamın özellikleri.

BİLGİSAYAR ÖZELLİKLERİ	AÇIKLAMA	PLATFORM
Merkezi İşlem Birimi (CPU)	Intel® Celeron(R)CPU 900 @ 2.20 GHz2.19GHz	MATLAB R2018'a
RAM	5	
Hard disk	107 GB	
İşletim sistemi	Windows 10 Home	

4.2. BSCA ile SVM hibrit algoritması için Parametre Ayarları

Tezde önerilen Binary SCA (BSCA)- SVM sisteminde uygulanabilmesi için öncelikle NSL-KDD99 ve UNSW-NB15 deneysel veri kümelerinin her ikisi de normalize edilmiş ve düzenlenmiştir. Daha sonra bu veri setlerinin her biri için deneyler gerçekleştirilmiş ve deney sonuçları değerlendirilmiştir. Tez kapsamında önerilen modelin sonuçları Çekirdek fonksiyonları (RBF çekirdekli RBF-SVM ve Polinom çekirdekli Polinom-SVM), İkili Parçacık Sürü Optimizasyonu entegreli SVM (BPSO-

SVM), Rastgele Orman (RF), K-En Yakın Komşuluk (k-NN), Karar Ağacı (DT), Naif Bayes (NB) ve literatürden seçilen bazı güncel mevcut algoritmalar ile karşılaştırılmıştır.

BSCA ile entegre edilmiş SVM sınıflandırıcısının parametrelerini ayarlayabilmek için, özellik boyutları [8-45] kapalı aralığında değiştirilerek tüm testler yapılmıştır, Agentsize Büyüklüğü ise 43 olmuştur. Algoritmanın maksimum iterasyon sayısı 100'dür. Uygunluk değerleri için eşitleme faktörü veya sabiti olarak adlandırılan A değeri 0.999 ve C constance değeri 2.0 olarak alınmıştır. Ayrıca, kıyaslamalarda SVM in yine sınıflandırıcı olarak kullanıldığı Parçacık Sürü Optimizasyonu parametreleri açısından; C1 ve C2 olarak belirtilen iki katsayının değeri 2.0 bulunmuştur. Populasyon Büyüklüğü ise 41 olmuştur. Çizelge 4.3'de tez kapsamında önerilen algoritmanın parametrelerinin değerleri gösterilmektedir.

Çizelge 4.3. Bu çalışmada uygulanan parametre değerleri.

BSCA + SVM		PSO + SVM	
Parametre	**Değer**	**Parametre**	**Değer**
AgentSize	43	Population Size	41
Maksimum iterasyon	100	Maksimum yineleme	100
Örnek veri büyüklüğü	100	Veri boyutunun belirlenmesi	100
Ortalama çalıştırılan iterasyon	40	Ortalama çalıştırılan iterasyon	40
A (eşitleme faktörü veya sabiti)	0.999	C1 Katsayı sabiti	2.0
C (Constance)	2	C2 Katsayı Sabiti	2.0

4.3. BSCA-SVM 'nin optimum Agentsize, Sigma ve C Parametre Değerlerini Bulma

Radyal temel fonksiyonu (Radial basic function (RBF)) çeşitli bilimsel problemleri çözmek için kullanılan Destek Vektör Makinesi'nin en önemli çekirdeklerinden biridir. Tez kapsamında C (Yumuşak Kenar Ceza Terimi) ve Sigma'nın parametre değerlerini elde etmek için arama uzayında C için 2^{-11} ile 2^{-7}, Sigma için ise 2^{-6} ile 2^{-1} arasındaki değerler öngörülmüştür. En yüksek doğruluk, C = 0.03125 ve Sigma = 0.000488 olduğunda elde edilmiştir.

Sinüs Kosinüs algoritması için en iyi ajan boyutunu (Agentsize) elde etmek amacıyla bu aralıkları Çizelge 4.4'de gösterildiği gibi 5 özellikten ve 50 özelliğe kadar

test edildi. Sinüs Kosinüs algoritmasının en iyi parametresini bulmak için 0.1'den 1'e kadar olan değerler dikkate alınmıştır. Parametre değerleri 0.999 olduğunda algoritmanın Çizelge 4.4'de tanımlanan optimum sonuçları elde etmesi dikkat çekici olmuştur.

Çizelge 4.4. NSL-KDD veriseti kullanılarak önerilen modelin optimum Agentsize değerinin bulunması.

Agentsize	Ölçüm Metrikleri sonuçları							
	Acc	Sen	Pre	Fm	Rec	Spe	FPR	EER
5 Ajan	92	100	80	88.9	100	89.3	0.11	0.075
10 Ajan	96	97.9	93	91.9	97	96.6	0.0344	0.05
15 Ajan	90	100	63.7	77.8	100	87.9	0.1212	0.1
20 Ajan	80	90	56.25	69.23	90	76.7	0.23	0.2
21 Ajan	95	91	91	91	91	96.6	0.035	0.05
25 Ajan	95	100	84.6	91.7	100	93.1	0.069	0.05
30 Ajan	95	100	77.8	87.5	100	93.94	0.06	0.05
35 Ajan	92.5	87.5	93.4	90.4	87.5	95.84	0.04	0.075
40 Ajan	82.5	100	56.3	72	100	96.6	0.035	0.02
41 Ajan	**99.01**	98.9	98	97	98.9	99.1	0.2	0.08
50 Ajan	92	90.9	83.33	86.9	91	93.10	0.069	0.075

Çizelge 4.4 de elde edilen en iyi Agentsize değerine göre Eşitleme Faktörü/Sabitinin olası değerleri [0.1-1.0] aralığında test edilmiş ve sonuçlar Çizelge 4.5'te gösterilmiştir.

Çizelge 4.5. NSL-KDD veriseti kullanılarak optimum eşitleme faktörü/sabiti parametresinin bulunması.

A	Ölçüm Metrikleri sonuçları (Agentsize=41 alınmıştır)							
Eşitleme Faktörü / Sabiti	Acc	Fm	Pre	Sen	Rec	Spe	FPR	EER
0.1	95.50	94.5	89.7	97.5	97.5	98.9	0.2	0.3
0.2	87.50	100	67	80	80	84	0.2	0.1
0.3	92	90.33	85.33	83	83	82	0.1	0.2
0.4	95	94	89	93	93	87	0.48	0.45
0.5	85	100	56	70	0	81	0.2	0.1
0.6	88	77	70	74	74	91	0.09	0.1
0.7	97	81	90	89	89	91.322	0.01	0.23
0.8	96.50	91	93	91	91	94	0.03	0.22
0.9	98.5	99	96	99.1	99.1	96.6	0.035	0.3
0.999	**99.30**	**97**	**98**	**98.9**	**98.9**	**99.1**	**0.2**	**0.08**
1	85	84.61	83.33	78.5	78.5	86	0.14	0.15

Önceki bölümde açıkladığımız gibi, bu çalışma için yine UNSW-NB15 veri kümesini kullandığımız yeni modelin performansını test etmek için iki saldırı veri

kümesi kullanılmıştır. UNSW-NB15, modern normal ve saldırı türlerini içeren, NSL-KDD gibi diğer veri kümelerinden daha fazla avantaja sahip olan yeni bir saldırı tespit veri setidir. Destek Vektör Makinesinin en potansiyel çekirdeklerinden biri olan Radyal temel fonksiyon (RBF) için en iyi parametreyi bulmak için benzer prosedür izlenir.

C ve Sigma parametre değerlerinde, C için 2^{-11} ile 2^{-7}, Sigma için 2^{-6} ile2^{-1} arasındaki değerler dikkate alınmıştır. Arama uzayında sigma için en yüksek doğruluk, C = 0.0078125 ve Sigma = 0.0625 parametresi olduğunda elde edilir. Çizelge 4.6'te tanımlanan Sinüs Kosinüs algoritmasının elde edilen sonuçları gösterilmiştir.

Çizelge 4.6. UNSW-NB15 veri setini kullanarak önerilen modelin optimum ajan boyutunu bulmak.

Agentsize	Ölçüm Metrikleri sonuçları							
	Acc	Fm	Pre	Sen	Rec	Spe	FPR	EER
5 Ajan	90	100	73	100	100	86.6	0.013	0.1
10 Ajan	77	69	64	66.7	66.7	81.5	0.18	0.23
15 Ajan	72.5	67	50	63.63	63.63	76	0.2	0.3
20 Ajan	92.50	89	86	92	92	93	0.23	0.2
21 Ajan	87	75	84	85	85	95	0.15	0.12
25 Ajan	95	90	98	89	89	83	0.2	0.25
30 Ajan	98	75	100	100	100	99.60	0.03	0.01
35 Ajan	90	75	78	100	100	86	0.14	0.1
40 Ajan	82.5	100	56.3	90	90	96.6	0.35	0.75
43 Ajan	**99.70**	**97.90**	**99.1**	**99.8**	**9.8**	**98**	**0.001**	**0.005**
50 Ajan	92	90.9	83.33	86.9	91	80	0.1	0.2

Çizelge 4.7. UNSW-NB15 veri setini kullanarak optimum parametrenin (eşitleme faktörü / sabit) elde edilmesi.

A	Ölçüm Metrikleri sonuçları (Agentsize=41 alınmıştır)							
Eşitleme Faktörü / Sabiti	Acc	Fm	Pre	Sen	Rec	Spe	FPR	EER
0.1	90	80	67	100	100	88	0.12	0.1
0.2	80	80	93	50	50	70	0.12	0.2
0.3	85	93	95	98	80	100	0.2	0.03
0.4	90	67	100	50	50	100	0.0	0.1
0.5	70	93	91	90	90	100	0.0	0.03
0.6	90	90	84	100	100	80	0.2	0.1
0.7	90	86	100	75	75	100	0.0	0.1
0.8	95	87.50	96	93	93	100	0.0	0.1
0.9	85	89	93	95	95	67	0.33	0.5
0.999	100	96	98	99	99	97	0.0	0.005
1	90	96	88	95	95	93.10	0.0	0.1

4.4. İkili SCA'nın RBF-SVM ile hibritleştirilmesi

Tezde geliştirilen algoritma her biri 100 iterasyon olmak üzere 40 defa çalıştırılmış ve ortalama sonuçlar elde edilmiştir. Deneylerde UNSW-NB15 ve NSL-KDD veri setleri kullanılmıştır. Önerilen algoritma UNSW-NB15 veri kümesine uygulandığında (c = 0.0625 ve sigma = 0.00390625 parametrelerinde) elde edilen doğruluk oranı %99.70 olmuştur. Önerilen algoritma NSL-KDD veri kümesine uygulandığında (c = 0.00048 ve sigma = 0.25 parametrelerinde) elde edilen doğruluk oranı %99.31 olmuştur. Karşılaştırma ve performans ölçütleri olarak seçilen Doğruluk (ACC), Duyarlılık (SEN), Özgüllük (SPE), Hassasiyet (PRE), F-skor (FM), Recall (Rec), Yanlış Pozitif Oran (FPR) ve Hata oranı (ERR) ölçüm sonuçları ve seçilen özellik sayısı Çizelge 4.8'de gösterilmiştir.

Çizelge 4.8. SCA nın RBFSVM ile hibritleştirilmesi sonucu elde edilen deney sonuçları.

SCA- RBFSVM		
	UNSW-NB15	NSL-KDD99
Acc	%99.70	%99.30
Sen	%99.80	%98.90
Pre	%99.01	%98.00
Fm	%97.90	%98.00
Rec	%99.80	%98.90
Spe	%98.00	%99.10
FPR	%0.001	%0.20
ERR	%0.005	%0.080
Seçilen özellik sayısı	8	11

4.5. İkili SCA'nın Polinomsal-SVM ile hibritleştirilmesi

Benzer veri seti ve aynı dağılım aralıkları kullanarak, SCA'nın PolinomSVM çekirdeği ile hibrit bir kombinasyonun veri setleri üzerindeki sonuçları elde edilmiştir. Çizelge 4.9'da ölçüm metrikleri cinsinden elde edilen performans sonuçları sunulmuştur.

Çizelge 4.9 SCA nın Polinomsal-SVM ile hibritleştirilmesi sonucu elde edilen deneysel sonuçlar.

SCA-POLİNOMSVM		
	UNSW-NB15	NSL-KDD99
Acc	%98.31	%97.00
Sen	%99.00	%96.00
Pre	%98.90	%98.50
Fm	%97.00	%95.00
Rec	%99.00	%96.00
Spe	%98.20	%98.80
FPR	%0.16	%0.50
ERR	0.08	0.06
Seçilen özellik sayısı	10	16

4.6. PSOnun SVM ile hibritleştirilmesi

Tezde önerilen BSCA-SVM yönteminin performansını karşılaştırmak için Parçacık Sürü Optimizasyonu (PSO) ile SVM entegrasyon edilmiştir. Sunulan yönteme uygulanan veri kümeleri, benzer şekilde BPSO-SVM üzerinde uygulanmış ve elde edilen sonuçlar Çizelge 4.10 Tablosunda gösterilmiştir.

Çizelge 4.10. BPSO-SVM'nin deneysel sonuçları.

BPSO-SVM		
	UNSW-NB15	NSL-KDD99
Acc	%99.31	%98.50
Sen	%98.90	%97.12
Pre	%97.50	%98.00
Fm	%99.00	%97.00
Rec	%98.90	%95.80
Spe	%99.15	%98.35
FPR	%0.30	%0.400
ERR	%0.16	%0.50
Seçilen özellik sayısı	14	16

4.7. Makine Öğrenme Algoritmalarının Deneysel Sonuçları

4.7.1. RBF çekirdeğinin deneysel sonucu

RBF, işlevsel verimliliği ve başarılı sonuçlara ulaşma yeteneği nedeniyle en çok tercih edilen ve en popüler Destek Vektör Makinesi çekirdeklerinden biridir. Bu çekirdek fonksiyonunu, çekirdek fonksiyonları arasında hangisinin iyi performans gösterdiğini görmek için önerilen yöntemi kullanarak aynı veri setleri üzerinde

karşılaştırdık. İki farklı veri seti NSL-KDD ve UNBW-NB15'in elde edilen sonuçları Çizelge 4.11'de gösterilmektedir.

Çizelge 4.11. RBF-SVM'nin deneysel sonuçları.

RBF-SVM		
	UNSW-NB15	NSL-KDD99
Acc	%80.00	%97.00
Sen	%61.54	%87.50
Pre	%70.37	%93.75
Fm	%61.54	%87.70
Rec	%50.50	%95.50
Spe	%55.17	%91.31
FPR	%0.325	%0.1000
ERR	%0.296	%0.016
Seçilen özellik sayısı	43	41

4.7.2. Polinomsal çekirdek fonksiyonunun deneysel sonucu

Polinomsal (homojen olmayan) çekirdek fonksiyonu, veri kümesi doğrusal olarak ayrılmadığında kullanılan Destek Vektörü Makine Çekirdeği fonksiyonlarından biridir. Tez kapsamında sunulan algoritmayı karşılaştırmak için bu çekirdek fonksiyonu da test edilmiştir. Polinomsal çekirdekli SVM'nin performansını önerilen algoritma ile karşılaştırmak için daha önce yaptığımız gibi benzer veri seti uygulanmıştır, ilk test NSL-KDD veri seti kullanılarak yapılmış ve ikincisin de ise UNBW-NB15 kullanılmıştır. Elde edilen sonuçlar Çizelge 4.12'de gösterilmiştir.

Çizelge 4.12. Polinomsal-SVM'nin deneyesel sonuçları.

Polinomsal-SVM		
	UNSW-NB15	NSL-KDD99
Acc	%85.00	%95.00
Sen	%50.00	%95.24
Spe	%82.36	%84.22
Pre	%55.33	%86.96
Rec	%50.00	%95.24
FM	%65.00	%90.91
ERR	%0.016	%0.23
FPR	%0.20	%0.100
Seçilen özellik Sayısı	43	41

4.7.3. Rasgele orman algoritmasının deneysel sonucu

Random Forest algoritmasının, bazı makine öğrenme algoritmalarına kıyasla çok güçlü ve etkili bir yapısı vardır. Sınıflandırma, regresyon ve hayatta kalma analizlerinde kullanılmaktadır. RF algoritmasını ayarlamak için tahmin edici (predictor) sayısı, ormandaki yetişkin ağaç sayısı ve bölme kuralları olmak üzere üç parametre gerekir. Yeni modelimizin performansını karşılaştırmak için RF algoritması da test edilmiştir. Çizelge 4.13'te elde edilen sonuçlar gösterilmiştir.

Çizelge 4.13. RF deneysel sonucu.

	RF	
	UNSW-NB15	NSL-KDD99
Acc	%99.00	%95.00
Sen	%98.00	%88.80
Spe	%99.00	%99.00
Rec	%98 .00	%88.80
Pre	%97.00	%98.90
FM	%98.9 0	%90.00
FPR	%0.100	%0.00
ERR	%0.09	%0.05
Seçilen Özellik Sayısı	43	41

4.7.4. K-en yakın komşuluk algoritmasının deneysel sonucu

K-en yakın komşuluk (K-nearest neighbors (k-NN)) algoritması, sınıflandırma ve regresyon için kullanılan parametrik olmayan ve benzerliğe dayalı denetimli bir öğrenme algoritmasıdır. k-NN sınıflandırmasının kuralı, örneklere en yakın k komşularını algılamak ve her bir eğitim verisini uygun sınıfa göre belirlemektir. Performansını aynı anda değerlendirmek için yeni sistemimizle k-NN karşılaştırılmıştır. Çizelge 4.14'te sonuçlar yer almaktadır.

Çizelge 4.14. K-en yakın komşuluk algorimasının deneysel sonuçları.

	k-NN	
	UNSW-NB15	NSL-KDD99
Acc	%98.00	%97.50
Sen	%90.91	%98.00
Spe	%99.00	%96.88
Pre	%98.89	%88.89
Rec	%90.91	%98.00
FM	%98.80	%94.12
FPR	%0.25	%0.025
ERR	%0.80	%0.031
Seçilen özellik Sayısı	43	41

4.7.5. Karar ağacının deneysel sonucu

Karar ağacı (Decision Tree (DT)) hem sınıflandırma hem de regresyon amaçları için kullanılan makine öğrenme algoritma tekniklerinden biridir. Bağımsız veri kümelerini eğitmek için bir karar ağacı üretilir, bu prosedür süreç sona erene kadar özyinelemeli olarak çalışır. Önerilen modelin karşılaştırılması için hem UNSW-NB15 hem de NSL-KDD veri setleri kullanılarak testler gerçekleştirilmiş ve Çizelge 4.15'te sonuçlar gösterilmiştir.

Çizelge 4.15. Karar ağacının deneysel sonuçları.

	k-NN	
	UNSW-NB15	NSL-KDD99
Acc	%98.00	%97.50
Sen	%90.91	%98.00
Spe	%99.00	%96.88
Pre	%98.89	%88.89
Rec	%90.91	%98.00
FM	%98.80	%94.12
FPR	%0.25	%0.025
ERR	%0.80	%0.031
Seçilen özellik Sayısı	43	41

4.7.6. Naif Bayes sınıflandırıcı deneysel sonucu

Naif bayes sınıflandırıcı, bağımsızlık özellikleri (varsayımlar) üzerinde oldukça etkili olan basit bir Bayes olasılık teoremidir. Her özelliğin, diğerlerini etkilemeden kendi olasılığı vardır. Naif bayes sınıflandırıcısının, önerilen model ile karşılaştırılmadan önce hem UNSW-NB15 hem de NSL-KDD veri setleri kullanılarak performansı değerlendirilmiştir. Çizelge 4.16'da sonuçlar gösterilmiştir.

Çizelge 4.16. Naive Bayes sınıflandırıcı deneysel sonuçları

	k-NN	
	UNSW-NB15	NSL-KDD99
Acc	%94.50	%91.90
Sen	%90.91	%81.82
Spe	%96.00	%96.00
Pre	%91.00	%88.00
Rec	%90.91	%81.82
FM	%95.23	%85.00
FPR	%0.050	%0.010
ERR	%0.01	%0.04
Seçilen özellik Sayısı	43	41

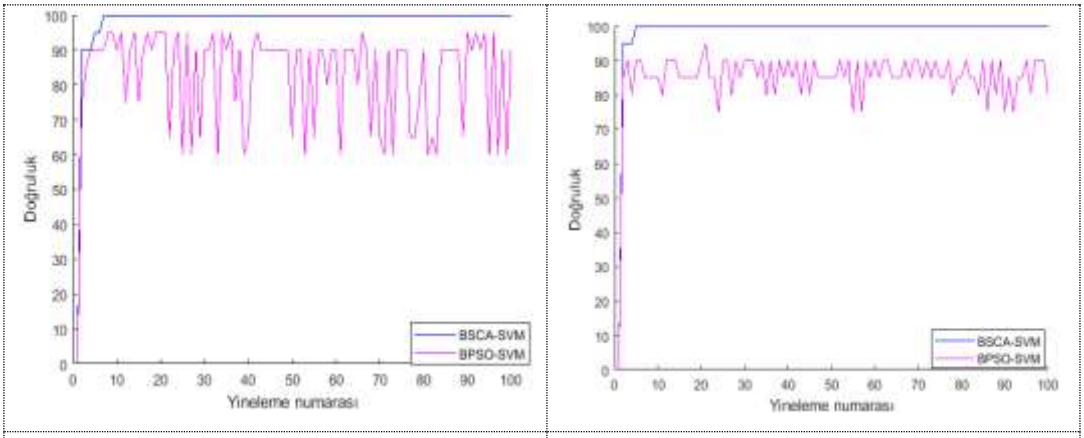
4.8. BPSO-SVM ve Önerilen Yöntemin (BSCA-SVM)'nin Karşılaştırılması

Bu bölümde, tezde önerilen algoritma olan Sinüs Kosinüs Algoritması ile entegre edilmiş Destek Vektör Makinası (BSCA-SVM) algoritması ve Parçacık Sürü Optimizasyonu ile entegre edilmiş Destek Vektör Makinesi (PSOSVM) algoritması sonuçları karşılaştırılmıştır. BSCA-SVM algoritması, BPSO-SVM algoritmasından daha üstün bir performans göstermiştir. Çizelge 4.17'deseçilen ölçüm metriklerine göre elde edilen karşılaştırma sonuçları sunulmuştur.

Çizelge 4.17. BPSO-SVM ve önerilen yöntemin karşılaştırılması (BSCA-SVM).

	BSCA-SVM		BPSO-SVM	
	UNSW-NB15	NSL-KDD99	UNSW-NB15	NSL-KDD99
Acc	**%99.70**	**%99.30**	%99.31	%98.50
Sen	%99.80	%98.90	%98.90	%97.12
Spe	**%99.01**	%98.00	%97.50	%98.00
Pre	**%97.90**	%**98.00**	%99.00	%97.00
Rec	%99.80	**%98.90**	%98.90	%95.80
FM	%98.00	%99.10	%99.15	%98.35
FPR	**%0.001**	**%0.20**	%0.30	%0.40
ERR	%**0.00**	**%0.08**	%0.16	%0.50
Seçilen özellik Sayısı	**8**	**11**	14	16

Doğruluk değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.1 de sunulmuştur. Şekil 4.1.a da NSL-KDD veri seti için iterasyona göre algoritmaların doğruluk değerleri değişimi grafiksel olarak verilmiştir. Şekil 4.1.b de ise UNSW-NB15 veri seti için iterasyona göre algoritmaların doğruluk değerleri değişimi grafiksel olarak sunulmuştur.

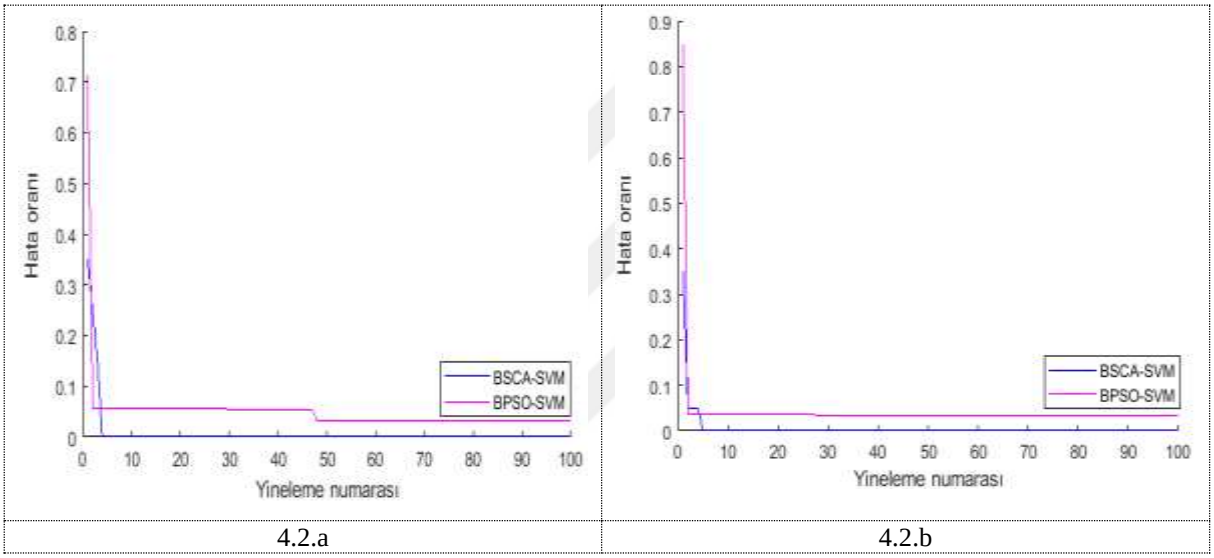


4.1.a 4.1.b

Şekil 4.1. BPSO-SVM ve BSCA-SVM algoritmalarının "Doğruluk" kriterine göre (4.1.a) NSL-KDD veri seti (4.1.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.1 incelendiğinde her iki algoritmanında çok hızlı yakınsadığı ve 10 iterasyona ulaşılmadan yüksek doğruluğa yaklaştıkları görülmektedir. Ama önerilen BSCA-SVM algoritması %100 doğruluğa ulaşabilmekteyken, BPSO-SVM algoritması %100 doğruluğa yaklaşamamıştır.

Hata Oranları değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.2 de sunulmuştur. Şekil 4.2.a da NSL-KDD veri seti için iterasyona göre algoritmaların doğruluk değerleri değişimi grafiksel olarak verilmiştir. Şekil 4.2.b de ise UNSW-NB15 veri seti için iterasyona göre algoritmaların doğruluk değerleri değişimi grafiksel olarak sunulmuştur.

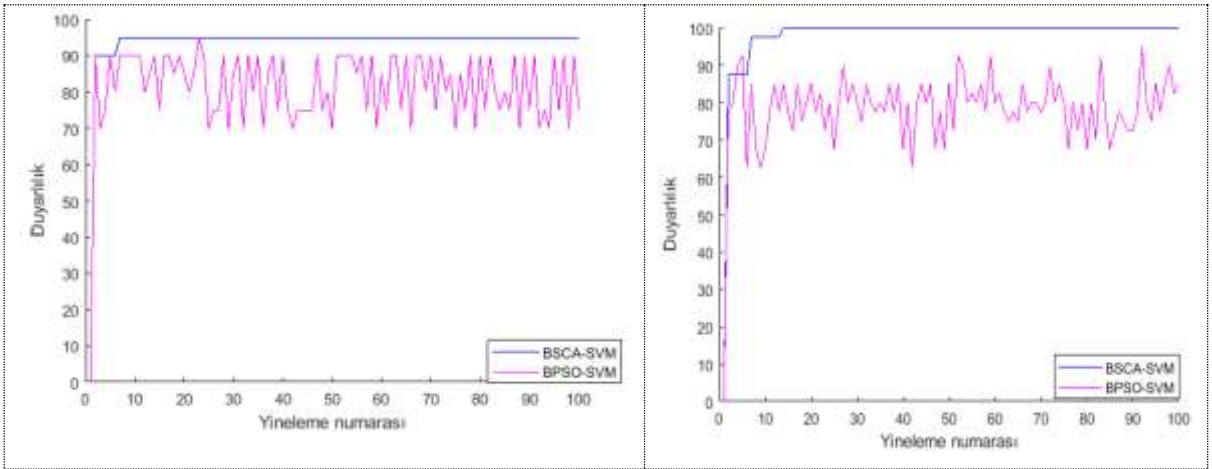


4.2.a 4.2.b

Şekil 4.2. BPSO-SVM ve BSCA-SVM algoritmalarının "Hata Oranı" kriterine göre (4.2.a) NSL-KDD veri seti (4.2.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.2 incelendiğinde, her iki algoritmanında %0 hata oranını ilk birkaç iterasyonda yakalayamadığı görülmektedir. İlk 5 iterasyondan sonra ise her iki veri seti için BSCA-SVM algoritmasının sıfır hata oranına inebildiği görülmektedir. BPSO-SVM ise 100 iterasyon sonunda bile sıfır hata oranına inememiştir.

Duyarlılık değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.3 de sunulmuştur.

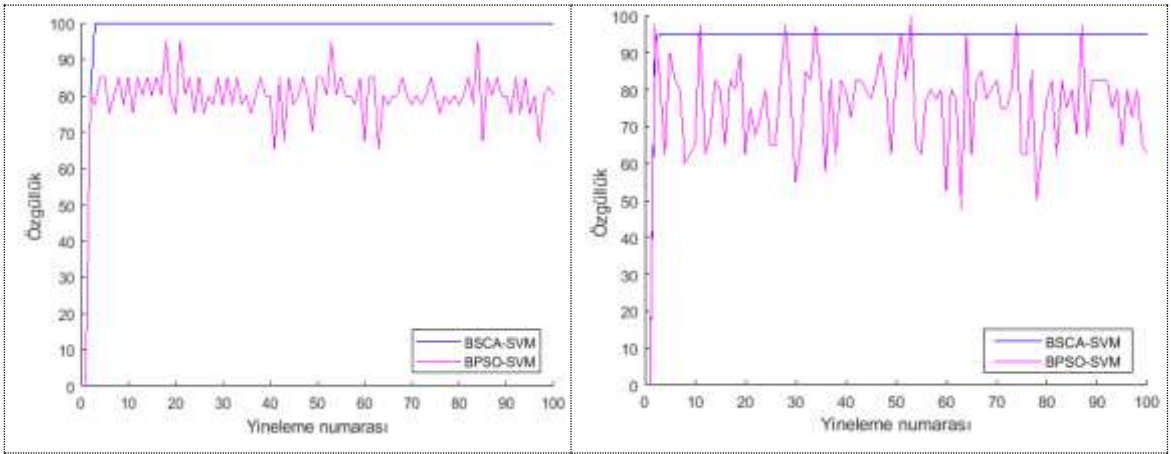


4.3.a 4.3.b

Şekil 4.3. BPSO-SVM ve BSCA-SVM algoritmalarının "Duyarlılık" kriterine göre (4.3.a) NSL-KDD veri seti (4.3.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.3'e göre, ilk grpah (4.3.a) Duyarlılık sonucu NSL-KDD veri seti kulanarak göre sunulan yontem (BSCA-SVM) BPSO-SVM'den daha iyi olduğunu göstermektedir. Aynı zamanda ikinci grpah'ın(4.3.b) Duyarlılık sonucu UNSW-NB15 veri seti kulanarak sunulan yontem (BSCA-SVM) BPSO-SVM'den daha başarlı olduğunu göstermektedir.

Özgüllük değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.4 de sunulmuştur.



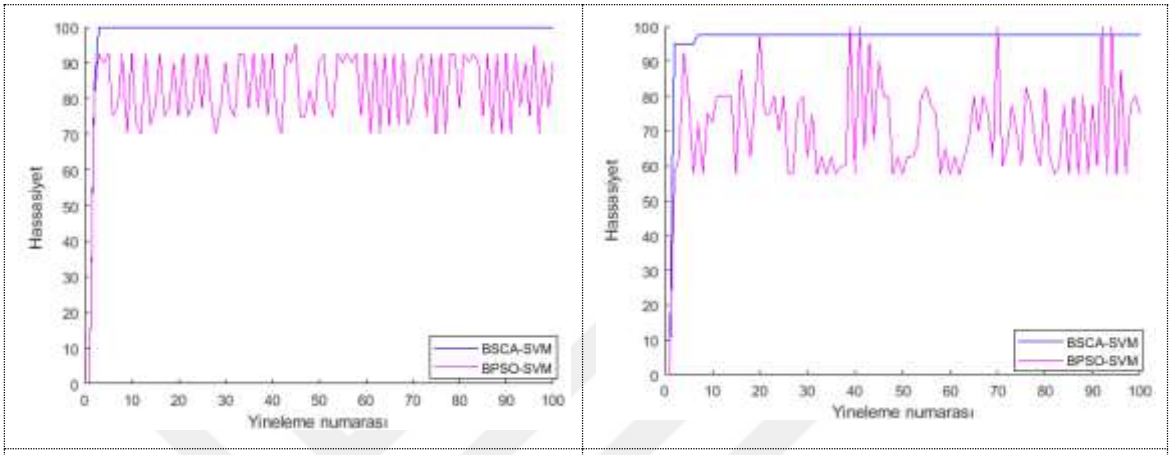
4.4.a 4.4.b

Şekil 4.4. BPSO-SVM ve BSCA-SVM algoritmalarının "Özgüllük" kriterine göre (4.4.a) NSL-KDD veri seti (4.4.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.4'e göre, birinci grpah'ın (4.4.a) Özgüllük sonucu NSL-KDD veri seti kulanarak göre sunulan yontem (BSCA-SVM) BPSO-SVM'den daha iyi olduğunu göstermektedir. fakat ikinci grpah'ın (4.4.b) Özgüllük sonucu UNSW-NB15 veri seti

kulanarak göre BPSO-SVM'nin sunulan yontemden daha iyi başarlı olduğunu göstermektedir.

Hassasiyet değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.5 de sunulmuştur.

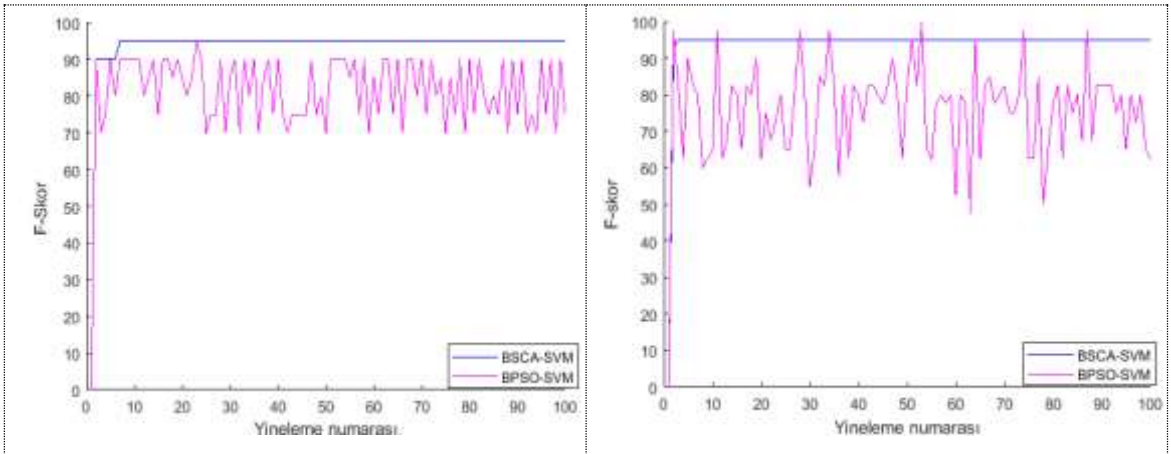


4.5.a 4.5.b

Şekil 4.5. BPSO-SVM ve BSCA-SVM algoritmalarının "Hassasiyet" kriterine göre (4.5.a) NSL-KDD veri seti (4.5.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.5'e göre ilk grpah'ın (4.5.a) Hassasiyet sonucu NSL-KDD veri seti kulanarak göre sunulan yontem (BSCA-SVM) BPSO-SVM'den daha başarılı olduğunu göstermektedir. İkinci paragraf (4.5.b) BPSO-SVM'nin sunulan yontemden daha iyi olduna göstermektedir.

F-Skor değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.6 de sunulmuştur.

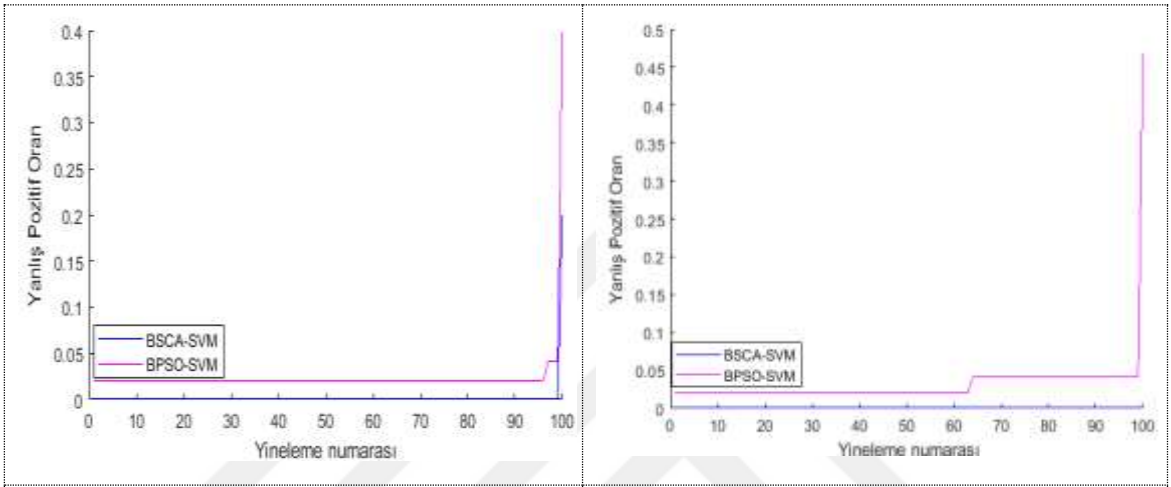


4.6.a 4.6.b

Şekil 4.6. BPSO-SVM ve BSCA-SVM algoritmalarının "F-skor" kriterine göre (4.6.a) NSL-KDD veri seti (4.6.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.6’e göre ilk grpah'ın (4.6.a) F-skor” sonucu NSL-KDD veri seti kulanarak göre sunulan yontem (BSCA-SVM) BPSO-SVM'den daha başarılı olduğunu göstermektedir. İkinci paragraf (4.6.b) göre BPSO-SVB'nin sunulan modelden daha başarılı olduğunu göstermektedir.

Yanlış Pozitif Oran değerine göre önerilen BSCA-SVM algoritması ve BPSO-SVM algoritmalarının karşılaştırmaları Şekil 4.7 de sunulmuştur.



4.7.a 4.7.b

Şekil 4.7. BPSO-SVM ve BSCA-SVM algoritmalarının “Yanlış Pozitif Oran” kriterine göre (4.7.a) NSL-KDD veri seti (4.7.b) UNSW-NB15 veri seti kullanılarak karşılaştırması.

Şekil 4.7'ye göre ilk şekil (4.7.a,) sunulan yöntemin (BSCA-SVM) sıfırdan başlayıp 0.2'ye yükseldiğini, BPSO-SVM'nin sıfırdan başlayıp 0.4'e yükseldiğini göstermektedir. Buda, sunulan yontem (BSCA-SVM) BPSO-SVM'den daha iyi olduğunu göstermektedir. Ikinci şekil (4.7.b) sunulan yöntemin (BSCA-SVM) sıfırdan başlayıp **0.001%**'ye yükseldiğini, BPSO-SVM'nin sıfırdan başlayıp 0.3'e yükseldiğini göstermektedir. Yanı sunulan yöntemin yeniden BPSO-SVM'den daha iyi olduğunu göstermektedir.

4.9. Önerilen Modelin Diğer Algoritmalarla Karşılaştırılması

Bu bölümde, aynı iki veri kümesi kullanılarak yapılan performans testlerine uygun şekilde tez kapsamında geliştirilen BSCA-SVM (İkili Sinüs Kosinüs Algoritması ile Destek Vektör Makinesi hibriti) algoritması ile farklı sınıflandırma yöntemleri karşılaştırılmıştır. Tezde geliştirilen BSCA-SVM modeli hem metasezgisel

algoritmalarla hem de makine öğrenme algoritmalarıyla karşılaştırılmıştır. Karşılaştırma için kullanılan yöntemler şunlardır; Klasik RBF çekirdekli Destek Vektör Makinesi (RBF-SVM), Klasik Polinom tabanlı Destek Vektör Makinesi (Polinom SVM), Rasgele Orman (RF), K-en yakın komşu (k-NN), Karar Ağacı (DT), Naïve Bayes Sınıflandırıcı (NB) ve ikili PSO ile entegre SVM (BPSO-SVM). Doğruluk metriği açısından her bir algoritma için elde edilen değerler Çizelge 4.18'de karşılaştırmalı olarak gösterilmiştir.

Çizelge 4.18. BSCA- SVM'nin doğruluk açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	**%99.70**	**%99.30**
BPSO-SVM	%99.31	%98.50
RBF SVM	%80.00	%97.00
Polinomsal SVM	%85.00	%95.00
RF	%99.00	%95.00
KNN	%98.00	%97.00
DT	%98.00	%97.50
NB	%94.50	%98.80

Sen, Spe, Pre, Fm, Fpr ve Err metrikleri açısından yapılan karşılaştırma tabloları sırasıyla Çizelge 4.18- Çizelge 4.24 de sunulmuştur

Çizelge 4.19. BSCA- SVM'nin duyarlılık açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	**%99.80**	**%98.90**
BPSO-SVM	%98.90	%97.12
RBF SVM	%61.54	%87.50
Polinomsal SVM	%50.00	%95.24
RF	%98%	%88.80
KNN	%90.91	%98.00
DT	%99.00	%98.90
NB	%90.91	%81.82

Çizelge 4.20. BSCA- SVM'nin özgüllük açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	%98.00	**%99.10**
BPSO-SVM	%99.15%	%98.35
RBF SVM	%70.37	%93.75
Polinomsal SVM	%82.36	%84.22
RF	%99.00	%99.00
KNN	%99.00	%96.88
DT	%98.00	%96.77
NB	%96.00	%96.00

Çizelge 4.21. BSCA- SVM'nin hassasiyet açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	**%99.01**	%98.00
BPSO-SVM	%97.50	%98.00
RBF SVM	%50.50	%95.50
Polinomsal SVM	%55.33	%86.96
RF	%97.00	%98.90
KNN	%98.89	%88.89
DT	%100.00	%95.25
NB	%91.00	%88.00

Çizelge 4.22. BSCA- svm'nin f-skor açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	%97.90	%**98.00**
BPSO-SVM	%99.00	%97.00
RBF SVM	%55.17	%91.31
Polinomsal SVM	%65.00	%90.91
RF	%98.90	%90.00
KNN	%98.80	%94.12
DT	%98.80	%95.00
NB	%95.23	%85.00

Çizelge 4.23. BSCA- SVM'nin yanlış pozitif oran açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	**%0.001**	%0.200
BPSO-SVM	%0.300	%0.400
RBF SVM	%0.296	%0.016
Polinomsal SVM	%0.200	%0.100
RF	%0.100	%0.00
KNN	%0.250	%0.025
DT	%0.0101	%0.020
NB	%0.0500	%0.010

Çizelge 4.24. BSCA- SVM'nin hata oranı açısından çeşitli algoritmalarla karşılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
BSCA-SVM	**%0.005**	%0.080
BPSO-SVM	%0.3000	%0.400
RBF SVM	%0.3250	%0.100
Polinomsal SVM	%0.0160	%0.230
RF	%0.0900	%0.050
KNN	%0.8000	%0.031
DT	%0.0160	%0.010
NB	%0.0100	%0.040

Karşılaştırma tablolarında (Çizelge 4.18- Çizelge 4.24) fark ettiğiniz gibi, önerilen metodu (BSCA-SVM) diğer algoritmalardan daha başarılıdır; örneğin, Çizelge 4.18 ve 4.19'daki doğruluk ve duyarlılık açısından diğerlerinden daha iyi görünmektedir. Buna

ek olarak, Çizelge 4.20 (özgüllük),4.22 (f-skor) NSL-KDD99 verileri kulanarak ve Çizelge 4.23 (pozitif oran) UNSW-NB15 kulanarak BSCA-SVM'nin hala diğerlerinden daha iyi olduğunu göstermektedir. Aksine, diğer algoritmalar daha iyidir.

4.10. Önerilen Model ile Mevcut Çalışmaların Karşılaştırılması

Bu bölümde, tez kapsamında geliştirilen BSCA-SVM yönteminin performansını görmek için, literatürden seçilmiş yeni bazı mevcut metasezgisel algoritmalar ile karşılaştırması yapılmıştır. Karşılaştırma için kullanılan yöntemler şunlardır; Akıllı Su Damlacıkları entegre SVM (Intelligent Water Drops integrated SVM(IWD-SVM)), Modifiye Edilmiş Binary Gri Kurt entegre SVM (modified binary grey wolf optimisation with SVM(MBGWO-SVM)), Aslan metasezgisel optimizasyonu ve Konvolüsyon Sinir Ağlarına (Lion optimization algorithm with Convolution Neural Networks (LOA+CNN)), Konvolüsyonlu Sinir Ağları ile Locust Swarm Optimizasyonu (LSO+FNN)), Parçacık Sürü Optimizasyonunun ile sinir ağları PSO + NN (particle swarm optimization Combined with neural Network), Yapay Arı Kolonisi ve Yapay Balık Sürsüsü Bulanık C-Means Kümeleme ve Korelasyona Dayalı Özellik Seçimi (Artificial Bee Colony and Artificial Fish Swarm algorithms with Fuzzy C-Means Clustering and Correlation-based Feature Selection (ABC & AFS + FCM&CFS)), Kaba Küme Teorisi ile SVM (Rough Set Theory and Support Vector Machine(RST+ SVM) , ki-kare özellik seçimi ve SVM (chi-square feature selectio and SVM), Beta Karışım Tekniğiile ile Anomali tespit sistemi (beta mixture technique with Anomaly detection system (BMM-ADS)) ,Çekirge Optimizasyon Algoritması ve Benzetilmiş Tavlama (Grasshopper Optimization algorithm and simulated annealing(GOSA).

Çizelge 4.25’de, sunulan model ile mevcut bazı çalışmalar arasındaki doğruluk açısından karşılaştırmalı deneysel sonuçlar verilmiştir. Çizelge 4.25'da doğruluk değerlerine göre önerilen algoritmanın diğer algoritmalara göre daha iyi olduğu gözlemlenebilmektedir.

Çizelge 4.25 NSL-KDD veri setini kullanarak BSCA-SVM'yi mevcut çalışmalarla kasılaştırılması.

Algoritmalar	UNSW-NB15	NSL-KDD99
IWD-SVM (Neha et al.2018)	%99.0915	9
GA-SVM (Aslahi-Shahri BM et al.2015)	%97.76	10
MBGWO-SVM (Qusay et al.2019)	%99.22	14
LOA-CNN (Dr. Arivudainam et al.2018)	%96.00	20
LSO-FNN (Ilyas et al.2019)	%94.04	Belirlenmemiş
PSO-NN (Ahmed et al.2019)	%98.10	12
ABC & AFS + FCM&CFS (Vajiheh et al.2018)	%99.00	6
RST-SVM (Chen et al.2009)	%89.13	29
Square FS-SVM (Sumaya et al.2017)	%98.00	31
BSCA-SVM (Önerilen yöntem)	**%99.30**	**11**

En başarısız yöntemin %89.13 ile RST-SVM metodu olduğu görülmektedir. En başarılı yöntem ise %99.30 ile tezde önerilen saldırı tespit yöntemidir. Kalan diğer yöntemlerin başarı sıralaması ise şu şekilde olmuştur: MBGWO-SVM, IWD-SVM, ABC&AFS+FCM&CFS, PSO-NN, Square FS-SVM, GA-SVM, LOA-CNN ve LSO-FNN.

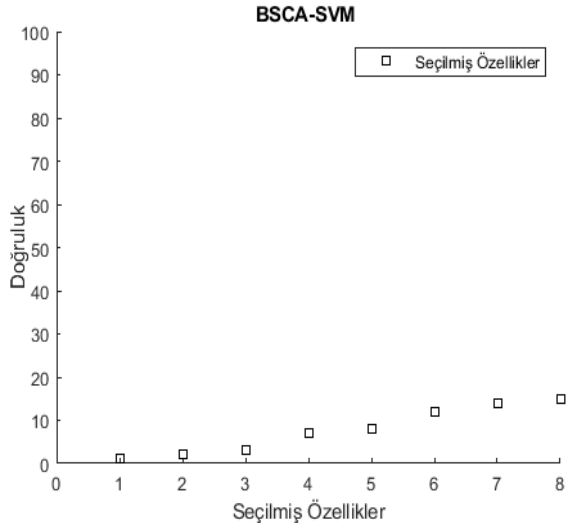
Çizelge 4.26'de, sunulan model ile mevcut bazı çalışmalar arasındaki doğruluk açısından karşılaştırmalı deneysel sonuçlar verilmiştir. Çizelge 4.26'da doğruluk değerlerine göre önerilen algoritmanın diğer algoritmalara göre daha iyi olduğu gözlemlenebilmektedir.

Çizelge 4.26 UNSW-NB15 veri setini kullanarak BSCA-SVM'yi mevcut çalışmalarla kasılaştırılması.

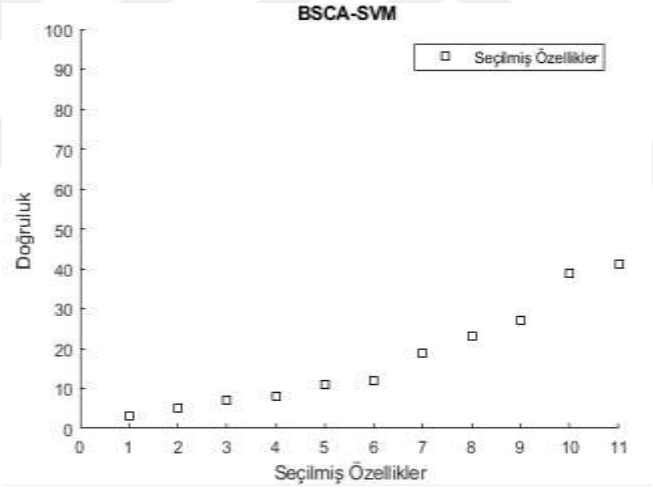
Algoritmalar	UNSW-NB15	NSL-KDD99
ABC & AFS + FCM&CFS (Vajiheh et al.2018)	98.9%	6
GOSA (Shubhra et al.2020)	98.96%	Belirlenmemiş
LSO-FNN (Ilyas et al.2019)	95.42%	Belirlenmemiş
BMM-ADS(N Moustafa et al.2018)	93.40%	8
BSCA-SVM (Önerilen yöntem)	**99.70%**	**8**

En başarısız yöntemin 93.40% ile BMM-ADS metodu olduğu görülmektedir. En başarılı yöntem ise %99,70 ile tezde önerilen saldırı tespit yöntemidir. GOSA ikinci sırada , ucuncu sirada ABC & AFS + FCM&CFS yer aldı, LSO-FNN dörduncu sirada yer almaktadir.

Şekil 4.8 ve 4.9'da, iki farklı saldırı veri kümesi kullanılarak BSCA-SVM algoritması ile seçilen özellikler gösterilmiştir.



Şekil 4.8. UNSW-NB15 veri kümesi için BSCA-SVM ile elde edilen özellik sayısının doğruluk değişim grafiği.



Şekil 4.9. NSL-KDD veri kümesi için BSCA-SVM ile elde edilen özellik sayısının doğruluk değişim grafiği.

Çizelge 4.27 ve 4.28'de, iki veri kümesi kullanılarak önerilen BSCA-SVM yöntemi tarafından seçilen özelliklerin ayrıntıları gösterilmektedir.

Çizelge 4.27. UNSW-NB15 veri kümesi için BSCA-SVM tarafından seçilen özellikler.

Seçilen özelliklerin sayısı	Özelliklerin adı	Özelliğin açıklaması
1	Id	Identification (integer)
2	Dur	Toplam süreyi kaydedin (Float).
3	XProt	İşlem protokolü(nominal)
5	XState	durum ve bağımlı protokolüne işaret eder, Örneğin. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN ve (-) (kullanılmıyorsa) (nominal).
10	Rate	Kaynak jitter (mSec) (Float)
20	Djit	Hedef jitter (mSec) (Float)
26	Synack	TCP bağlantısı kurulum süresi, SUN ve SYN_ACK paketleri arasındaki süre. (Float)
35	ct_src_dport_ltm	Son servise göre 100 bağlantıda aynı hizmeti (14) ve kaynak adresini (1) içeren bağlantı sayısı. (26). (Integer)

Çizelge 4.28. NSL-KDD veri kümesi için BSCA-SVM tarafından seçilen özellikler.

Seçilen özelliklerin sayısı	Özelliklerin adı	Özelliğin açıklaması
1	Duration	Bağlantı zamanı (sec.)
7	Land	Bağlantının kaynağı ve hedefi aynı sunucu / bağlantı noktası 1, diğer 0 ise (Sürekli)
18	num_52s	Açık deniz hayvanı kabuğu sayısı
23	Count	Aynı sunucuya t sn cinsinden yapılan bağlantı sayısı (Continuous)
26	sr3_serror_rate	“SYNC” hatası ile alınan bağlantıların yüzdesi (Continuous)
30	same_sr3_rate	Aynı servise yapılan bağlantıların yüzdesi (Continuous).
32	sr3_diff_host_rate	Farklı sunuculara yapılan bağlantıların yüzdesi (Continuous).
35	dst_host_same_sr3_rate	Farklı sunuculara yapılan bağlantıların yüzdesi (Continuous)
37	dst_host_same_src_port_rate	Aynı kaynak bağlantı noktasından yapılan bağlantıların yüzdesi (Continuous)
40	dst_host_sr3_serror_rate	“SYN” hatasıyla alınan bağlantıların yüzdesi (Continuous)
41	dst_host_sr3_rerror_rate	"REJ" hatası ile alınan bağlantıların yüzdesi (Continuous)

5. SONUÇLAR VE TARTIŞMA

Bu tezde, saldırı tespiti sistemlerinde yüksek performans elde etmek ve aynı zamanda doğruluğu artırmak için makine öğrenme algoritmalarından seçilen algoritmaların entegre edilmesi ve hibritleştirilmesi hedeflenmiştir. Tez kapsamında, özellik seçimindeki başarısından dolayı BSCA nın özellik seçimi için kullanılması ve sınıflandırmadaki başarısından dolayı da SVM algoritmasının kullanılması öngörülmüştür. Tez çalışmasında iyi bilinen diğer klasik makine öğrenmesi tekniklerinin de kullanımları ve başarı oranları dikkate alınmıştır. Bu tez çalışmasının amacı, makine öğrenmesi algoritmalarından seçilen bir kısmının saldırganları tespit etme potansiyelini vurgulamak ve bir optimum hibrit saldırı tespit sistemi elde etmektir. Tercih edilecek hibrit algoritmanın performansını değerlendirmek için NSL-KDD ve UNSW-NB15 veri setleri kullanılmıştır. Ayrıca, geliştirilen BSCA-SVM modelin RBF ve Polinom çekirdekli Destek Vektör Makinesi (RBF-SVM ve Polinom SVM), Rastgele Orman (RF), K-en yakın komşu (k-NN), Naive Bayes sınıflandırıcısı (NB), İkili Parçacık Sürü Optimizasyonu entegreli SVM (BPSO-SVM) ve literatürden seçilen güncel mevcut bazı çalışmalarla karşılaştırmaları (Akıllı Su Damlacıkları entegre SVM (Intelligent Water Drops integrated SVM(IWD-SVM)), Modifiye Edilmiş Binary Gri Kurt entegre SVM (modified binary grey wolf optimisation with SVM(MBGWO-SVM)), Aslan metasezgisel optimizasyonu ve Konvolüsyon Sinir Ağlarına (Lion optimization algorithm with Convolution Neural Networks (LOA+CNN)), Konvolüsyonlu Sinir Ağları ile Locust Swarm Optimizasyonu (LSO+FNN)), Parçacık Sürü Optimizasyonunun ile sinir ağları PSO + NN (particle swarm optimization Combined with neural Network), Yapay Arı Kolonisi ve Yapay Balık Sürsüsü Bulanık C-Means Kümeleme ve Korelasyona Dayalı Özellik Seçimi (Artificial Bee Colony and Artificial Fish Swarm algorithms with Fuzzy C-Means Clustering and Correlation-based Feature Selection (ABC & AFS + FCM&CFS)), Kaba Küme Teorisi ile SVM (Rough Set Theory and Support Vector Machine(RST+ SVM) , ki-kare özellik seçimi ve SVM (chi-square feature selectio and SVM), Beta Karışım Tekniğiile ile Anomali tespit sistemi (beta mixture technique with Anomaly detection system (BMM-ADS)) ,Çekirge Optimizasyon Algoritması ve Benzetilmiş Tavlama (Grasshopper Optimization algorithm and simulated annealing(GOSA)) yapılmıştır.

Bu çalışmada, Sinüs Kosinüs algoritması önemli özellikleri seçmek ve genel olarak özellikleri azaltmak için kullanılmış, sınıflandırma için ise Destek Vektör

Makinesi tercih edilmiştir. Tüm popülasyon çözümleri ikili dize formundadır. Her ikili dize bir ajanı temsil etmektedir. Ajanın ikili dizeleri, mevcut değerine göre azar azar ilerletilir, bu yaklaşım optimum bit değerlerinin bulunmasına yardımcı olur. Sunulan yeni yöntemin (BSCA-SVM) performansını değerlendirmek için UNSW-NB15 ve NSL-KDD olmak üzere iki saldırı veri seti kullanılmıştır ve bu verisetlerinde doğruluk değerleri sırasıyla %99,70 ve %99,30 olarak elde edilmiştir. Sınıflandırma; SVM (RBF ve Polinom), RF, k-NN, DT, NB ve BPSO-SVM gibi modellere göre daha başarılı sonuç vermiştir.

Bu tezin hedefi, saldırganları verimli bir şekilde tespit edebilen ve değerlendirme performansında üstün bir çözüme ulaşabilen optimum yeni bir hibrit saldırı tespit modeli elde etmektir. Ek olarak, önerilen sistem, yeni modern saldırganları tanıma ve başarılı bir şekilde davetsiz misafirleri önleme yeteneğine sahiptir. Modelin hem UNSW-NB15 hemde NSL-KDD veri setleri üzerindeki performansı incelendiğinde, alternatif olarak kullanılabilecek yeni bir yöntem olduğu görülebilmektedir. Literatürdeki çalışmalara yarışabilir düzeyde doğruluk değerlerine ulaşılmıştır. Performansın daha iyi analiz edilebilmesi için yedi farklı kıyaslama metriği de kullanılmıştır.

KAYNAKLAR

Jasmin Kevric et al., 216, An effective combining classifier approach using tree algorithms for network intrusion detection, *Neural Comput & Applic, DOI 10.1007/s00521-016-2418-1*.

Neha Acharya and Shailendra Singh ., 2017, An IWD-based feature selection method for intrusion detection system, *Soft Comput.DOI 10.1007/s00500-017-2635-2*.

Arif Jamal Malik1 & Farrukh Aslam Khan., 2017, a hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection, *Springer Science+ Business Media, LLC. DOI 10.1007/s10586-017-0971-8, 2017.*

Vajiheh Hajisalem & Shahram., 2018, A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, *Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran. Elsevier, 2018.*

B. M. Aslahi-Shahri & et al., 2015, A hybrid method consisting of GA and SVM for intrusion detection system, *Neural Comput & Applic (2016) 27:1669–1676 DOI 10.1007/s00521-015-1964-2*.

Asmaa Shaker Ashoor & Prof. Sharad Gore., 2011, Importance of Intrusion Detection system (IDS)", *International Journal of Scientific & Engineering Research, Volume 2, Issue 1, January-2011 1 ISSN 2229-5518.*

Salma Elhag & et al., 2011, A multi-objective evolutionary fuzzy system to obtain a broad and accurate set of solutions in intrusion detection systems, *Springer-Verlag GmbH Germany 2017, Soft Comput DOI 10.1007/s00500-017-2856-4.*

James Anderson, J. P., 1980, Computer security threat monitoring and surveillance, *James P. Anderson Co.Box 42 Fort Washington, Pa. 19034215 646-4706.*

Alheeti, 2011, Intrusion Detection System and Artificial Intelligent, *Alanbar University, Iraq, https://www.researchgate.net/publication/221911298.*

Oktay and O.K. Sahingoz., 2013, Attack Types and Intrusion Detection Systems in Cloud Computing, *6th International information security & cryptology conference*.

Hafez et al., 2016, Sine Cosine Optimization Algorithm for Feature Selection, *ScientificResearchGroupinEgypt(SRGE), http://www.egyptscience.net,2016 IEEE* .

Mohamed et al., 2017, A Hybrid Method of Sine Cosine Algorithm and Differential Evolution for Feature Selection, *Springer International Publishing AG 2017, DOI: 10.1007/978-3-319-70139-4_15.*

Issa et al., 2018, ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment, *2018 Elsevier Ltd. All rights reserved.*

Navid and Saheb., 2015, Comb ination of PSO Algorithm and Nai ve Bayesian Classification for Parkinson Disease Diagnosis, *ACSIJ Advances in Computer Science: an International Journal, Vol. 4, Issue 4, No.16 , July 2015,ISSN : 2322-5157.*

Manikandan and Bhuvaneswari., 2017, An intelligent intrusion detection system for secure wireless communication using IPSO and negative selection classifier, *Springer Science+Business Media,LLC,part of Springer Nature 2018.*

Saqr and Santosh., 2017, Addressing Challenges for Intrusion Detection System using Naive Bayes and PCA Algorithm, *2017 2nd International Conference for Convergence in Technology (I2CT), ©2017 IEEE.*

Bahareh et al., 2014, Intrusion Detection System in Computer Network Using Hybrid Algorithms (SVM and ABC), *Journal of Advances in Computer Research Quarterly pISSN: 2345-606x eISSN: 2345-6078 Sari Branch, Islamic Azad University, Sari, I.R.Iran (Vol. 5, No. 4, November 2014), Pages: 43-52 www.jacr.iausari.ac.ir .*

Dewan et al., 2011, Adaptive Intrusion Detection based on Boosting and Naïve Bayesian Classifier, *International Journal of Computer Applications (0975 – 8887)Volume 24– No.3, June 2011.*

Tom M.Mitchell., 1997, Machine learning, *Publisher: Science/Engineering/Math; (March 1, 1997), I SBN: 0070428077.*

Ethem Alpaydın., 2010, Introduction to Machine Learning Second Edition, *2010 Massachusetts Institute of Technology.*

Patrick ve ark., 2014, An Overview of Machine Learning with SAS ® Enterprise Miner™", *Patrick Hall, Jared Dean, Ilknur Kaynar Kabul, Jorge Silva SAS Institute Inc.*

Tsai et al., 2009, Intrusion detection by machine learning*: A review , 2009 Elsevier Ltd. All rights reserved.doi:10.1016/j.eswa.2009.05.029*

Saurabh and Sharma., 2014, Intrusion Detection using Naive Bayes Classifier with Feature Reduction", *Department of Computer Science, Banasthali University, Jaipur,Rajasthan, 304022,India.*

Amjad et al., 2018, NBC-MAIDS: Naïve Bayesian classification techniquev in multi-agent system-enriched IDS for securing IoT against DDoS attacks, *Springer Science+Business Media, LLC, part of Springer Nature 2018.*

Levent et al., 2012, A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier, *Department of Engineering Management and Systems*

Engineering at The George Washington University, Washington, DC, USA,2012 Elsevier Ltd. All rights reserved.

Mrutyunjaya Panda and Manas Ranjan Patra., 2007, Network intrusion detection using naïve bayes, *IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.12, December 2007*.

Khalil El-Khatib and Member,. 2010, Impact of Feature Reduction on the Efficiency of Wireless Intrusion Detection Systems", *IEEE Transactions on Parallel and DISTRIBUTED SYSTEMS, VOL. 21, NO. 8, AUGUST 2010, Khalil El-Khatib, Member, IEEE.*

Seyedali Mirjalili et al., 2018**,** SCA: A Sine Cosine Algorithm for solving optimization problems,*2015 Elsevier B.V. All rights reserved.*

K Srikanth et al., 2017, A New Binary Variant of Sine–Cosine Algorithm: Development and Application to Solve Profit-Based Unit Commitment Problem,, *Received: 16 May 2017 / Accepted: 1 August 2017© King Fahd University of Petroleum & Minerals 2017.*

Moustafa and Jill., 2015, A hybrid feature selection for network intrusion detection systems: Central points, *DOI:10.4225/75/57a84d4fbefbb Originally published in the Proceedings of the 16th Australian Information Warfare Conference (pp. 5-13).*

Nour Moustafa and Jill,. 2015 , UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems (UNSW-NB15 network data set), *Publisher: IEEE, DOI: 10.1109/MilCIS.2015.7348942", https://ieeexplore.ieee.org/abstract/document/7348942*

Arvinder et al., 2017, Hybridization of K-Means and Firefly Algorithm for intrusion detection system, *Spring, Int J Syst Assur Eng Manag, DOI 10.1007/s13198-017-0683-8.*

M A Jabbar et al., 2017, RFAODE: A Novel Ensemble Intrusion Detection System, *1877-0509 © 2017 The Authors. Published by Elsevier B.V.*

Shailendra and Sanjay., 2009, A Survey of Cyber Attack Detection Systems, *IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.5, May 2009.*

Huiwen Wang et al., 2017, An effective intrusion detection framework based on SVM with feature augmentation. *Elsevier B.V. All rights reserved.*

Rana et al., 2016, Toward an efficient fuzziness based instance selection methodology for intrusion detection system, *Spring,int. j. mach. learn. & cyber. (2017) 8:1767–1776,doi 10.1007/s13042-016-0557-4.*

R. Sindhu et al., 2007, Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism, *Springer 2017, Neural Comput & Applic DOI 10.1007/s00521-017-2837-7.*

A M VISWA BHARATHY and A MAHABUB BASHA., 2016, A multi-class classification MCLP model with particle swarm optimization for network intrusion detection", *Sa dhana ,Indian Academy of Sciences, DOI 10.1007/s12046-017-0626-8*

Dr. Eduard Belitser and Valeria Fonti., 2017, Feature Selection using LASSO,*University of Amsterdam.*

Rizk M. Rizk-Allah et al., 2018, Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems, Elsevier.*Faculty of Engineering, Department of Engineering Mathematics, Minufiya University, Egypt.*

Chong-zhi et al., 2018, Privacy-preserving Naive Bayes classifiers secure against the substitution-then-comparison attack, *Elsevier Inc. All rights reservedElsevier Inc. All rights reserved.*

Chih-Fong, Yu-Feng, Chia-Ying ,and Wei-Yang., 2009, Intrusion detection by machine learning: A review, *Elsevier Ltd. All rights reserved. journal homepage: www.elsevier.com/locate/eswa.*

Belavagi and Balachandra., 2016, Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection, *Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license.*

Srinivas Mukkamala, Guadalupe Janoski and Andrew Sung., 2002**,** Intrusion Detection Using Neural Networks and Support Vector Machines, *Srinivas Mukkamala, Guadalupe Janoski, Andrew Sung{ srinivas, silfalco, sung} @cs.nmt.edu Department of Computer Science New Mexico Institute of Mining and Technology Socorro New Mexico, 87801 USA. 2002 IEEE.*

Farid, ve ark., 2004, Classification of Hyperspectral Remote SensingImages With Support Vector Machines, *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 42, NO. 8, AUGUST 2004.*

Chiwen ve ark., 2018, A Modified Sine-Cosine Algorithm Based on Neighborhood Search and Greedy Levy Mutation", *Hindawi.Computational Intelligence and Neuroscience sVolume 2018, Article ID 4231647, 19 pages https://doi.org/10.1155/2018/4231647.*

Ekiz ve ark., 2017, Solving Constrained Optimization Problems with Sine-Cosine Algorithm, *Duzce, Turkey University, Department of Computer Engineering, Periodicals of Engineering and Natural Scinces ISSN 2303-4521Vol.5, No.3,* DOI: 10.21533/pen.v5i3.131. *pp. 378~386 Available online at: http://pen.ius.edu.ba.*

Vijay and Dinesh., 2018, Data Clustering Using Sine Cosine Algorithm:Data Clustering Using SCA, *Published in the United States of America bycIGI GlobalInformation Science Reference (an imprint of IGI Global) 701 E. Chocolate AvenueHershey PA, USA 17033, Chapter · April 2017 DOI: 10.4018/978-1-5225-2229-4.ch031. See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/315831525.*

M. Ali Aydın et al.,2009, A hybrid intrusion detection system design for computer network security, *Elsevier Ltd. All rights reserved, Department of Computer Engineering, Faculty of Engineering, Istanbul University, 34320 Avcilar, Istanbul, Turkey.*

Hari and Aritra., 2012, A Hybrid System for Reducing the False Alarm Rate of Anomaly Intrusion Detection System, *IEEE, 1 st Int'l Conf. on Recent Advances in Information Technology | RAIT-2012|.*

Paliwal and Ravindra., 2012, Denial-of-Service, Probing & Remote to User (R2L) Attack Detection using Genetic Algorithm, *International Journal of Computer Applications (0975 – 8887)Volume 60– No.19, December 2012.*

Naveen and Sihagç., 2018, A Novel Adaptive Sine Cosine Algorithm for Global Numerical Optimization, *International Journal of Innovative Science and Research Technology*
ISSN No:-2456 –2165, Volume 3, Issue 2, February – 2018.

I. Rish et al., 2014, An empirical study of the naive Bayes classifier, *T.J. Watson Research Center, 30 Saw Mill River Road,Hawthorne, NY 10532. Phone +1 (914) 784-7431.*

Babaoğlu ve ark., 2010**,** A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine, *Expert Systems with Applications 37 (2010) 3177–3183. Department of Computer Engineering, Selcuk University, Konya, Turkey. 2009 Elsevier Ltd. All rights reserved.*

Juan ark., 2018, Modified Naive Bayes Algorithm for network Intrusion Detection based on Artificial Bee Colony Algorithm, *The 4th IEEE International Symposium Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems 20-21 September,2018,Lviv,Ukriane.©2018 IEEE.*

James Kennedy and Russell Eberhart., 1995, Particle Swarm Optimization,*Washington, DC 20212, Purdue School of Engineering and Technology,© 1995 IEEE.*

James Kennedy and Russell Eberhart., 1997, A DISCRETE BINARY VERSION OF THE PARTICLE SWARM ALGORITHM, *Bureau of Labor Statistics Washington, DC 20212, Purdue School of Engineering and Technology,© 1997 IEEE.*

L.Dhanabal 1 and Dr. S.P. Shantharajah., 2015, A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms, *International Journal of Advanced Research in Computer and Communication Engineering,Vol. 4, Issue 6, June 2015, DOI 10.17148/IJARCCE.2015.4696.*

Tong Li et al., 2018, Differentially private Naive Bayes learning over multiple data sources, *Elsevier Inc. All rights reserved, 0020-0255/© 2018.*

Riccardo Poli James Kennedy and Tim Blackwell,. 2007, Particle swarm optimization An overview , *Published online: 1 August 2007,© Springer Science + Business Media, LLC 200, Received: 19 December 2006 / Accepted: 10 May 2007.*

Russell C. Eberhart &Yuhui Shi., 2001, Particle Swarm Optimization:Developments, Applications and Resources, 0-7803-6657-3/01/ 02001 IEEE.

Tamer F. Ghanem et al ., 2014, A hybrid approach for efficient anomaly detection using metaheuristic methods, *Production and hosting by Elsevier B.V. on behalf of Cairo University.Elsevier 2014.*

Gauthama Raman M R et al., 2017, A Hypergraph and Arithmetic Residue-based Probabilistic Neural Network for classification in intrusion detection systems, *.PII:S0893-6080(17)30033-3,DOI: http://dx.doi.org/10.1016/j.neunet.2017.01.012 Reference: NN 3715.*

Yuhui Shi and Russell Eberhart , 1998, A Modified Particle Swarm Optimizer, *0-7803-4869-9198 .0001998 EEE.*

Ozgur Depren et al., 2005, An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks, *Elsevier Ltd. All rights reserved. Bogazici University, Electrical and Electronics Engineering Department, Information and Communications Security (BUICS) Lab, Bebek, Istanbul, Turkey.*

Allen et al., 2000. Russell Eberhart (1998), A New Optimizer Using Particle Swarm Theory , *Sixth International Symposium on Micro Machine and Human Science,0-7803-2676-8/95 01995 IEEE.*

Justin Y. Lee and Mark P. Styczynskiç, 2018, NS-kNN:a modified k-nearest neighbors approach for imputing metabolomics data, *Received: 6 August 2018 / Accepted: 15 November 2018 / Published online: 23 November 2018© Springer Science+Business Media, LLC, part of Springer Nature 2018.*

Samuel E. Buttrey and Ciril Karo., 2002**,** Using k-nearest-neighbor classi&cation in the leaves of a tree, *Received: Elsevier Science B.V. All rights reserved. PII: S0167-9473(01)00098-6, Department of Operational Research OR=Sb, Naval Postgraduate School, Monterey CA 93943, USA.*

J.R. QUINLAN et al ., 1986, Induction of Decision TreesMachine Learning ,*1: 81-106,1986 Kluwer Academic Publishers, Boston - Manufactured in The Netherlands. (Received August 1, 1985)***.**

Antonio Paeza et al., 2018, Inducing non-orthogonal and non-linear decision boundaries in decision trees via interactive basis functions, *Elsevier Ltd. All rights reservedExpert Systems With Applications 122 (2019) 183–206.https://doi.org/10.1016/j.eswa.2018.12.041*.

Yasufumi Ochiai a et al , 2018, Improvement of time table robustness by analysis of drivers' operation based on decision trees, *2-17-1 Tsudanuma, Narashino, Chiba, 275-0015, Japa..Received 3 April 2018; Received in revised form 4 March 2019; Accepted 5 March 2019∗ © 2 0 1 9 P u b l i s h e d b y E l s e v i e r L t d.*

Lev V.Utkina et al., 2019, A weighted random survival forest, *Elsevier B.V. All rights reserved. https://doi.org/10.1016/j.knosys.2019.04.015,0950-7051*

Yangming Zhou,GuopingQiub.,2018**,** Random forest for label ranking, *ElsevierLtd.Allrights reserved. https://doi.org/10.1016/j.eswa.2018.06.036,0957-*

Hathiram et al.,2017, Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking, *PII: S1568-4946(17)30578-1. DOI: https://doi.org/10.1016/j.asoc.2017.09.039. Reference: ASOC 4486.*

Sahlol and Ewees., 2016,Training Feedforward Neural Networks Using Sine-Cosine Algorithm to Improve the Prediction of Liver Enzymes on Fish Farmed on Nano-selenite, *Faculty of Specific Education, Damietta University Damietta, Egypt.*

Mohamed et al., 2017, An improve d Opposition-Base d Sine Cosine Algorithm for global optimization, *Elsevier Ltd. All rights reserved.*

Sai Li, Huajing Fang and Xiaoyong Liu., 2017, Parameter Optimization of Support Vector Regression Based on Sine Cosine Algorithm, *PII: S0957-4174(17)30583-3, DOI: 10.1016/j.eswa.2017.08.038, Reference: ESWA 11507.*

Qusay et al., 2019, Intrusion detection system based on a modified binary grey wolf optimization, *received: 23 July 2018/Accepted: 15 February 2019. Neural Computing and Applicationshttps://doi.org/10.1007/s00521-019-04103-1.*

D. Arivudainambi1 et al., 2019**,** LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks, received: 5 January 2018 / Accepted: 21 February 2018_ The Natural Computing Applications Forum 2018.

Ilyas Benmessahel et al., 2019, A new evolutionary neural networks based on intrusion detection systems using locust swarm optimization, *Received: 12 April 2018 / Revised: 24 September 2018 / Accepted: 31 December 2018 © Springer-Verlag GmbH Germany, part of Springer Nature 2019, Evolutionary Intelligence https://doi.org/10.1007/s12065-019-00199-5.*

Sydney and Yanxıa., 2019, A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection System,
Received March 6, 2019, accepted March 12, 2019, date of publication March 18, 2019, date of current version April 5, 2019.Digital Object Identifier 10.1109/ACCESS.2019.2905633.

Ahmad Shokoohsaljooghi and Hamid Mirvaziri., 2019, Performance improvement of intrusion detection system using neural networks and particle swarm optimization algorithms,
Received: 11 August 2017/Accepted: 2 May 2019, Bharati Vidyapeeth's Institute of Computer Applications and Management 2019. Published spring online: 20 May 2019.

Sumaya and Aswani, 2019, Intrusion detection model using fusion of chi-square feature selection and multi class SVM,
Received 7 July 2015; revised 4 October 2015; accepted 3 December 2015 Available online 31 March 2016.

Aslahi-Shahri BM et al., 2015, A hybrid method consisting of GA and SVM for intrusion detection system. Neural Comput Appl. doi:10.1007/s00521-015-1964-2 Chen R-C, Cheng K-F, Hsieh C-F (2009).

Hajisalem V and Babaie S., 2018, A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection. Comput Netw 136:37–50

Karami A and Guerrero-Zapata M ., 2015, A hybrid multiobjective RBF-PSO method for mitigating DOS attacks in named data networking. Neurocomputing 151:1262–1282

N Moustafa et al., 2018, Anomaly detection system using beta mixture models and outlier detection. In: Pattnaik P, Rautaray S, Das H, Nayak J (eds) Progress in computing, analytics and networking. Advances in intelligent systems and computing, vol 710. Springer, Singapore, pp 125–135

Shubhra et al., 2020 Incorporating evolutionary computation for securing wireless network against cyberthreats. pringer Science+Business Media, LLC, part of Springer Nature 2020.