ANALYSIS OF NETWORK SECURITY USING MACHINE LEARNING METHODS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ANKARA UNIVERSITY

by

Maryam SALATİ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR PHYLOSOPHY IN
COMPUTER ENGINEERING

ANKARA
2024

ABSTRACT

Ph. D. Thesis

ANALYSIS OF NETWORK SECURITY USING MACHINE LEARNING METHODS

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Prof. Dr. İman ASKERBEYLİ

In this study, a novel approach for intrusion detection using a metaheuristic-based feature selection method combined with convolutional neural networks (CNNs). The main idea behind this method is to find the most effective features in a database and reuse them in the final layers of CNN architectures to increase the accuracy. There are several datasets that are commonly used for cyber security intrusion detection research and development. We used in this research 3 datasets .NSL-KDD, DEFCON and CDX datasets each has 41, 10, and 874 features, respectively.

The proposed method in this thesis is includes 4 stages. The main idea behind this method is to find the most effective features in a database and reuse them in the final layers of CNN architectures to increase the accuracy. The four stages are data pre-processing, pre-training, training and testing. The final goal is to train a CNN model to be used in order to detect intrusion in real time. The feature selection method employs a decision tree and a metaheuristic algorithm to select the most important features from different datasets. These tests are conducted for three meta-heuristic algorithms including GWO, MOPSO, and NSGA-II. The selected features are then fed into CNNs, including ResNet50, VGG16, and Efficient Net, to improve the accuracy of intrusion detection.

Experimental results on several benchmark datasets show that the proposed method can be promising in terms of different criteria. The proposed method is suitable for online and real-time intrusion detection as feature selection is performed during the pre-training phase. The findings of this study demonstrate the potential of the proposed method to effectively identify and classify intrusions in network traffic.

January 2024, 132 pages

Key Words: Intrusion detection, Convolutional Neural Network, Meta-heuristic algorithms, Feature selection, Decision Tree.

ÖZET

Doktora Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK AĞ GÜVENİRLİĞİ ANALİZİ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İman ASKERBEYLİ

Bu çalışmada, evrişimli sinir ağları (CNN'ler) ile birleştirilmiş metasezgisel tabanlı bir öznitelik seçim yöntemi kullanan izinsiz giriş tespiti için yeni bir yaklaşım. Bu yöntemin arkasındaki ana fikir, bir veritabanındaki en etkili özellikleri bulmak ve doğruluğu artırmak için bunları CNN mimarilerinin son katmanlarında yeniden kullanmaktır. Siber güvenlik saldırı tespiti araştırma ve geliştirmesi için yaygın olarak kullanılan birkaç veri seti vardır. Bu araştırmada 3 veri seti kullandık. NSL-KDD, DEFCON ve CDX veri setleri sırasıyla 41, 10 ve 874 özellik içeriyor. Siber saldırılar, kişisel, finansal ve resmi bilgilerimizin giderek daha fazla çevrimiçi olarak saklandığı günümüz dünyasında artan bir endişe kaynağıdır. İzinsiz Giriş Tespit Sistemi (IDS), kullanıcı kimlik doğrulamasını destekleyerek, güvenli erişim sağlayarak gizlilik kaybını önlemektedir.

IDS bilgisayar ağlarını saldırılardan korumayı hedeflediğinden, bilgisayar ve ağ güvenliğinin kritik bir yönüdür. Bir IDS'nin işlevi, verileri toplamak, analiz etmek ve daha sonra ek inceleme için bir insan ağ analistine iletilen uyarılar oluşturmak için bir algılama mekanizmasına dayanmaktadır . İnternetin ve iletişimin hızlı büyümesi iletilen verilerde büyük bir artışa neden olmuştur. Saldırganlar bu verilere göz dikmekle, çalmak veya bozmak için sürekli olarak yeni saldırılar oluşturmaktadırlar. Bu saldırıların artması, sistemlerin güvenliği için bir sorundur ve izinsiz giriş tespiti için en büyük zorluklardan birini meydana getirmektedir. IDS, ağ trafiğini inceleyerek izinsiz girişleri tespit etmeye yardımcı olan bir araçtır. Birçok araştırmacı yeni IDS çözümleri üzerinde çalışmış ve bu çözümleri oluşturmuş olsa da, yanlış alarm oranlarını azaltırken iyi bir algılama doğruluğuna sahip olmak için IDS'nin hala iyileştirilmesi gerekmektedir. Ek olarak, birçok IDS sıfırıncı gün saldırılarını tespit etmekte zorlanmaktadır. Son zamanlarda, bu alanda çalışılan makine öğrenimi (Machine Learning, ML) algoritmaları, ağ izinsiz girişini verimli bir şekilde ve yüksek doğrulukla tespit etmek için yapılan araştırmacılar arasında popüler bir çalışma alanı olarak öne çıkmaktadır.

Kural tabanlı yöntemler, basit ve yürütülmesi hızlı olmakla birlikte, eksik veya gürültülü verileri telafi edemez ve güncellenmesi zordur. Bu sorunların üstesinden gelmek için, kesin olmayan bilgilerin işlenmesini sağlamak için istatistik temelli

yaklaşımlar önerilmiştir; bununla birlikte, bu tür yöntemler yüksek bir hesaplama maliyeti gerektirir ve büyük miktarlarda veriyi işlemek için sınırlı bir yeteneğe sahiptir. Son zamanlarda, ML tabanlı yaklaşımlar üzerinde karmaşık izinsiz giriş modellerini tespit etmek için büyük miktarda veri üzerinde eğitilebilen karmaşık çıkarım modellerini kullanma yetenekleri nedeniyle giderek daha fazla çalışılmaktadır. Yeni ağ paradigmalarının ve karmaşık çıkarım modellerinin ortaya çıkmasına yol açan, internet üzerinden iletilen artan veri miktarı nedeniyle, bu t siber güvenlik ve IDS'lere yönelik makine öğrenimi tabanlı yaklaşımlara odaklanılmıştır. bu tezde izinsiz ağ girişlerini, anormallikleri ve diğer saldırı türlerini tespit etmek için CNN'nin çeşitli kullanımları önerilmiştir.

Önerilen yöntem, birkaç veri seti kullanılarak değerlendirilmiş ve elde olunan sonuçlar, çeşitli siber tehdit türlerini tespitinde etkinliğini göstermiştir. Geleneksel izinsiz giriş tespit sistemleri genellikle, etkinlikleri sınırlı olabilen ve hızla gelişen tehditlere ayak uydurmak için mücadele edebilen kural tabanlı yaklaşımlara veya imza tabanlı yöntemlere güvenirler. Bu sınırlamaların üstesinden gelmek için araştırmacılar, izinsiz giriş tespitinin doğruluğunu ve verimliliğini artırmak için makine öğrenimi tekniklerinin kullanımı üzerine geniş araştırmalar yapmışlardır.

Elde edilen sonuçların, ilgili yöntemin doğruluk, duyarlılık, F1 skor ve özgüllük açısından diğer son teknoloji yöntemlerden daha iyi performans gösterdiği kanıtlanmıştır. Bu iş izinsiz giriş tespitini iyileştirme yaklaşımımızın potansiyelini vurgulamaktadır. Saldırı tespit sistemleri, bilgisayar ağlarının güvenliğinin sağlanmasında kritik bir rol oynamaktadır. IDS oluşturmaya yönelik yaygın bir yaklaşım, özellik seçimi ve derin öğrenme algoritmaları gibi makine öğrenimi tekniklerini kullanmaktır. Yapılan çalışmalarda IDS'ler için çeşitli özellik seçim yöntemleri önerilmiştir. Bu yöntemler, hesaplama maliyetlerini en aza indirirken izinsiz girişleri tespit etmek için en uygun olan bir özellik alt kümesini tanımlamayı amaçlamaktadırlar.

Metasezgisel algoritmalar, izinsiz giriş tespit sistemlerinde özellik seçimi için kullanılabilen bir optimizasyon algoritmaları ailesidir. Özellikle GA’lar bu bağlamda yaygın olarak kullanılmaktadır. Benzetimli tavlama (simulated annealing) ve tabu arama gibi diğer metasezgisel algoritmalar da araştırılmıştır . Bu yöntemler, geniş bir arama alanında en uygun özellik alt kümesini aramak için esnek ve ölçeklenebilir yol önermektedirler. Bu tezde önerilen yöntem 4 aşamadan oluşmaktadır. Bu yöntemin arkasındaki ana fikir, bir veritabanındaki en etkili özellikleri bulmak ve doğruluğu artırmak için bunları CNN mimarilerinin son katmanlarında yeniden kullanmaktır. Dört aşama veri ön işleme, ön eğitim, eğitim ve testtir. Nihai amaç, saldırıyı gerçek zamanlı olarak tespit etmek için kullanılacak bir CNN modeli yetiştirmektir. Özellik seçme yöntemi, farklı veri kümelerinden en önemli özellikleri seçmek için bir karar ağacı ve metasezgisel bir algoritma kullanır. Bu testler, GWO, MOPSO ve NSGA-II dahil olmak üzere üç meta-sezgisel algoritma için yürütülür. Seçilen özellikler daha sonra izinsiz giriş tespitinin doğruluğunu iyileştirmek için ResNet50, VGG16 ve Efficient Net dahil olmak üzere CNN'lere beslenir.

Birkaç kıyaslama veri setindeki deneysel sonuçlar, önerilen yöntemin farklı kriterler açısından umut verici olabileceğini göstermektedir. Ön eğitim aşamasında özellik seçimi yapıldığından, önerilen yöntem çevrimiçi ve gerçek zamanlı saldırı tespiti için

uygundur. Bu çalışmanın bulguları, önerilen yöntemin ağ trafiğindeki izinsiz girişleri etkili bir şekilde tanımlama ve sınıflandırma potansiyelini göstermektedir.

Ocak 2024, 132 sayfa

Anahtar Kelimeler: İzinsiz giriş tespiti, Evrişimli Sinir Ağı, Meta-sezgisel algoritmalar, Özellik seçimi, Karar Ağacı.

ACKNOWLEDGEMENT

I want to express my gratitude to Allah for His assistance in successfully accomplishing this research. Furthermore, I extend appreciation to Prof. Dr. İman ASKERBEYLİ, for helping and collaboration of this PhD thesis. Finally, I can proudly declare that I have completed this endeavor.

I also would like to thank Assoc.Prof. Dr. Mehmet Serdar GÜZEL, Prof. Dr. Ali BOZBEY, Assoc.Prof. Dr. Gazi Erkan BOSTANCI and Prof.Dr. Ihsan Tolga Medeni for their guidance in my project.

Maryam SALATI
Ankara, January 2024

TABLE OF CONTENTS

SYMBOLS AND ABBREVIATIONS

y_i	Target (output) value
$\hat{y}_i$	output model's i-th scalar value
Σ_{ii}	covariance matrix i-th diagonal component
AP_i^{gen}	awareness probability of the j-th crow at gen
AP_{min} and AP_{max}	minimum and maximum values of the AP
FL_i^{gen}	flight length of the i-th crow at gen
$I_b^{(L)}$	Input of FNN network layer L
I_r	index set of the neurons
I_s	index set of the third layer neurons
K_{RBF}	FWRVM's radial basis function
L_i	set of principle components
$O_b^{(L)}$	Output of FNN network layer L
X_S	Covariance matrix X
c_q	Center of fuzzy membership function
g_2	variable obtained from the interval [0, 1]
m_i^{gen}	Memory of each crow
$p_i \in \mathbb{R}^m$	column vectors
$rank_i$	individual rank of i-th crow after sorting
w_{BF}	best feasible weights
w_{max} and w_{min}	maximum and minimum values of the weight coefficient
w_{sr}	weight coefficients between the fourth and third layers
x_i^{gen}	position of the individual i-th crow at any generation
α_i and α_j	random numbers with uniform distribution in [0,1]
λ_m	diagonal elements
σ_q	fuzzy membership function's width
B	diagonal matrix
max $_gen$	maximum number of generations
N	samples
P(p|X)	probability to the likelihood of bread instance
$\Lambda = diag(\lambda_1, \lambda_2, \dots, \lambda_m)$	diagonal matrix containing the Eigen values in decreasing order
Φ	$N \times (N+1)$ policy matrix
CUM	cumulative sum
FN	False Negative
FP	False Positive
I	Identity matrix
NP	size of the population
$P(w\|p, S, \alpha)$	weight posterior probability
$P(w\|\alpha)$	marginal probability
$P \in \mathbb{R}^{m \times m}$	Probability matrix

TN	True Negative
TP	True Positive
$X = [x_1^T, x_2^T, \dots, x_n^T]^T \in \mathbb{R}^{n \times m}$	data matrix
$X = \{x_1, x_2, \dots., x_n\}$	training dataset
$gbest$	optimal solution during current generation
p, q, r, s, t	neurons in FNN layers
$s = [s_1, s_2, \dots, s_N]$	fuzzy membership vectors
$w(t)$	weight coefficient
$w = (w_0, w_2, \dots, w_N)^T$	parameters that are adaptable
$y(x)$	estimated decision
$\alpha = [\alpha_0, \alpha_1, \dots, \alpha_N]$	vector of $N + 1$ hyperparameters
ε	distance between the selected i-th crow and the following j-th crow
$\sigma(y)$	Logical sigmoid function
β	The standardized coefficients of logistic regression
°C	Celsius
%	Percentage
g	Gram
l	Liter

Abbreviations

ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Networks
AV	p-Anisidine value
BN	Bayesian Network
BPNN	Back Propagation Neural Network
CARS	Competitive Adaptive Reweighted Sampling
CART	Classification and Regression Trees
CFNN	Cascade Forward Neural Network
CNN	Convolution Neural Network
CSA	Crow Search Algorithm
DAP	dynamic awareness probability
DNN	Deep Neural Networks
DSTARS	Deep Structure for Tracking Asynchronous Regressor Stack
DWT	Discrete Wavelet Transform
ELM	Extreme Learning Machines
ERD	Ensemble of Regressor Chains
EVOO	Extra Virgin Olive Oil
FAHP	fuzzy hierarchical analysis
FCF	Fermented Chickpea Flour
FD-VS	Fisher Discriminant-Variable Selection
FF	flesh firmness
FFNN	Feed Forward Neural Network
FIS	Fuzzy Inference System
FLM	Fuzzy Logic Modeling
FN	False Negative

FNN	Fuzzy Neural Network
FP	False Positive
FWRVM	Fuzzy Weighted Relevance Vector Machine
GA	Genetic Algorithm
GBR	Gradient Boosting Regressor
GENOPT-SVM	Genetic Algorithm for Optimization of SVM
GG	Guar Gum
GLCM	Gray Level Co-occurrence Matrix
GNB	Gaussian Naive Bayes
GPR	Gaussian Process Regression
GSF	grape seed flour
HPMC	Hydroxyl-Propyl Methyl-Cellulose
ICSA	Improved Crow Search Algorithm
IPCA	Improved PCA
IRIV	iteratively retaining informative variables
KDD	Knowledge Discovering Databases
KNN	K-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LIBS	Laser Induced Breakdown Spectra
LOO	Lampante Olive Oil
LR	Linear Regression
LSE	Least Square Estimation
LSSVM	Least Square SVM
LSSVR	Least Squares Support Vector Regression
MATLAB	Matrix Laboratory
MF	Median Filtering
MIMO	Multiple Input and Multiple Output
MISO	Multiple Input and Single Output
ML	Machine Learning
MLP	Multilayer Perceptron
MLR	Multivariate Linear Regression
MLR	Multiple Linear Regression
M-RNN	Mask Recurrent Neural Network
MSC	multiplicative scattering correction
MSE	Mean Square Error
MSI	multispectral imaging
MSSE	mean of sum squares error
MT	Multi Target
MTRS	Multi-target Regressor Stacking
NB	Naive Bayes
NEAT	Neuro-Evolution of Augmenting Topologies
NIRS	Near-Infrared Spectroscopy
NN	Neural Networks
N-PLS	N-way PLS
PCA	Principal Component Analysis
PCR	Principal Component Regression

PLS	Partial Least Squares
PLS-DA	Partial Square Discriminant Analysis
PSO	Particle Swam Optimization
PV	Peroxide Value
QP	quadratic programming
RBF	radial basis function
RELU	Rectified Linear Unit
RF	Random Forest
RMSEP	Root Mean Square Error of Prediction
RNN	Recurrent Neural Network
RPD	Ratio of Prediction to Deviation
RR	Ridge Regression
RVM	Relevance Vector Machine
SGDC	Stochastic Gradient Decision Classifier
Si-PLS	Synergy interval Partial Least Squares
SL	Shelf Life
SMOTE	Synthetic Minority Oversampling Technique
SMP	skim milk powder
SNV	standard normal variant
SSC	soluble solids content
SSC/TA	ratio of soluble solids to titratable acidity
SSE	sum squares error
SVM	Support Vector Machine
SVMR	SVM regression
TN	True Negative
TOTOX	total oxidation value
TP	True Positive
TSK	Takagi-Sugeno-Kang
UV	Ultraviolet
VIS	visible spectroscopy
VOO	Virgin Olive Oils
WF	wheat flour
WHO	World Health Organization
WQI	Water Quality Index
XG	Xanthan Gum
XGB	Extreme gradient boosting

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Overview

Today, The Internet holds paramount significance in human society and has directly affected a large part of human life. Many of the things that used to be done manually are now done online in the shortest time possible. There are extensive examples in this regard. Some of the applications of the Internet are: banking, online shopping, virtual education, messaging and many other applications. Undoubtedly, the Internet has gained a high level of influence among communities today. Although the Internet has many benefits, its use requires caution. The Internet also has its problems. One of the major problems with the Internet is its security. Endangering Internet security can have dangerous consequences. Intrusion represents a major challenge for the security of the Internet. Therefore, researchers are trying to establish Internet security with the help of various methods, one of which is machine learning and classification tools.

Therefore, in this research, an attempt has been made to provide a machine learning-based method for detecting intrusion into Internet networks.

1.2 Expression of the Problem

Most organizations and companies face the challenge of cyber security (Singh and Banerjee 2020). Computer security and cyberspace are known for their three characteristics of confidentiality, integrity and accessibility (Sah and Banerjee 2020). Intrusion detection is a vital component of cyber security. Intrusion detection systems are primarily based on two main groups: signature-based and anomaly-based (Alzubi et al. 2019). Anomaly-based intrusion detection systems form a pattern of normal network traffic and detect traffic that deviates from the constructed pattern as intrusion. In these systems, unknown attacks are detected but have a high positive error rate (Dwivedi et al. 2019). Machine learning-based approaches have gained much attention to develop more automated and accurate models (Jasim et al. 2022). One of the primary obstacles

in constructing intrusion detection systems using machine learning is to improve and accurately classify intrusion and reduce false positives (Miah et al. 2019). Learning algorithms can be categorized in to three types including supervised learning, unsupervised learning and semi-supervisory learning. The main problem of intrusion detection systems is working with large and complete data that the data set has multidimensional features and countless attributes (Sah and Banerjee 2020). Feature selection is a solution to fortify the effectiveness of intrusion detection systems by eliminating irrelevant and redundant features (Zhou et al. 2020). Random forest is a group classifier that performs better than other classifiers and is also an effective algorithm for reducing traits (Farnaaz and Jabbar 2016). Support vector machine represents a powerful model to perform linear and nonlinear classification. Our research relies on the trustworthy (NSL-KDD) dataset to teach and test proposed methods (other datasets are provided in the Tools and Methods section). The NSL-KDD dataset has forty-two features and five distinct classes, normal class & 4 intrusion classes U2R[1], R2L[2], DOS[3], and Probe (Singh and Venkatesan 2018). The proposed method uses meta-heuristic algorithms to select features and different machine learning algorithms for classification.

1.3 Computer Networks

A computer network is a collection of several computers and peripherals that are arranged together for the purpose of shared use. Depending on the layout of the computers connected to the network, each user can use the information on their computer and other computers on the network. Computer networking aims at the following purposes: (Pombeiro and Silva 2015):

Sharing resources: Sharing an information resource or computer peripherals, regardless of the geographical location of each resource, is called sharing resources.

[1] user-to-root
[2] Remote to Local
[3] Disk Operating System

Cost reduction: Concentrating resources and sharing them and avoiding distributing them in different units and the specific use of each user is an effective approach for cost reduction.

Reliability: This feature in networks indicates the presence of backup servers in the network, indicating the possibility to make backup copies of various information sources and systems in the network, and if one of the information sources in the network is not available." Used backups "Due to system failure". Support for servers in the network increases the efficiency, performance and constant readiness of the system.

Reduce time: Another goal of building computer networks is to create strong connections between remote users; That is, there is no exchange of information without geographical restrictions. This reduces the time it takes to exchange information and use resources automatically.

Scalability: development of local area network doesn't necessarily require changing whole the system; implying that there is no extra cost than that of facilities applied used for network expansion.

Communication: Users can exchange messages through existing innovations such as e-mail or other information systems; it is even possible to transfer files.

1.3.1 Internet

The Internet is an immense and intricate database that has become indispensable in the information age. It is virtually impossible to envision a world without it, highlighting the significance of discussing its importance. The majority of our daily tasks, communication, and sources of entertainment are heavily reliant on the Internet. In its essence, the Internet serves as a bridge connecting users across various mobile devices and computer systems. It offers a vast social platform where individuals can freely express their ideas and utilize them in diverse ways. An internet connection is essential

for the sharing and exchanging of ideas, information, and news. Moreover, Internet serves as a global network linking computers, businesses, people, government initiatives, and the captivating stories that unfold worldwide.

The presence of a high level of interconnected nodes on the Internet makes it possible for unauthorized access to other people's information. Therefore, it is necessary to provide specific measures to prevent cyber attacks on the Internet. Currently, communication and computer networks are an inevitable part of information technology topics. Whenever network and communication are mentioned, network security is also mentioned along with it. (Pombeiro and Silva 2015).

1.3.2 Network security

Network security represents a set of practices taken to prevent security challenges in the network. This set of measures can be implemented as several solutions in the form of hardware and software services. Indeed, network security is the process of preventing hardware and software measures to protect network resources against unauthorized access by anonymous individuals who intend to steal or destroy information is a network traffic monitoring and analysis system designed to identify potentially malicious activities. Detection & reporting of anomalies and suspicious activity is a key feature of IDS systems.

Cyber analytics can be categorized in three major divisions *viz.* abuse-Based (signature-Based), anomaly-based abnormality and hybrid (Jasim et al. 2022).

The purpose of implementing Abuse-Based techniques is to detect known attacks based on their signatures. Although Abuse-Based technique performs well in detecting known attacks, it requires several manual updates and is unable to detect new (zero) attacks.

Anomaly-based techniques are based on remembering network behavior and detecting anomalies as a potential attack. An anomaly may indicate an intrusion. In fact, in this

method, the analyzer looks for unusual cases. To do this, the collected information is examined to find items that indicate unusual actions. In Anomaly-based, intrusions will be identified by comparing observed current events with predetermined events. In this way, the characteristics of the collected events are considered based on a specific pattern and then compared with the new event. A major drawback of anomaly-based methods is the potential for high FN[4] false negative rates, as previously neglected system behaviors may be classified as irregularities.

Combination techniques leverage unconventional and unorthodox methods to enhance the detection rate of known intrusions and minimize the false positive (FP) rate when dealing with unknown attacks. By utilizing a combination of approaches, these techniques aim to improve the overall effectiveness of intrusion detection systems. They integrate multiple detection mechanisms, such as signature-based detection, anomaly detection, behavior analysis, and machine learning algorithms, to achieve a more comprehensive and robust defense against diverse threats. This amalgamation of techniques helps in increasing the accuracy of identifying known attacks while reducing the chances of false alarms or misclassifications when encountering previously unseen or novel attack patterns. The synergy created through combination techniques provides a powerful tool for enhancing the security posture and resilience of systems against intrusions and cyber threats.

An in-depth study of the literature did not reveal many methods of detecting pure anomaly; Most of the methods were really hybrid. Therefore, in describing ML[5] and DM[6] methods, anomalous detection methods and combined methods are described together.

The other part is based on the fact that they are looking for unwanted behavior: network-based or host-based. A network-based neural network detects intrusions through traffic control through network devices. Host-based IDS examines the process and activities associated with a particular host-related software environment.

[4] False Negative
[5] Machine Learning
[6] Data Mining

1.3.3 Importance of network security

If we understand the importance of information networks and its essential role in future social perception, the importance of the security of these networks becomes clear. If network security is not established, its many benefits will not be well realized, and money and e-commerce, customer service, personal information, public information, and electronic publications are all vulnerable to manipulation and material and intellectual abuse. Manipulation of information as the intellectual foundation of nations by internationally organized groups is also considered as a kind of disruption of national security and aggression against states and a national threat. For our country, where many basic software, such as the operating system, application software and the Internet, are provided through foreign intermediaries and companies, there is a fear of infiltration through secret channels. In the future, as banks and many other institutions and devices operate through the network, preventing malicious agents from infiltrating the network will become a strategic issue that, if not addressed, will cause damages that are sometimes irreparable. If a special message, for example from Microsoft, is sent to all Iranian sites and the operating systems in response to this message destroy and disable the systems, what huge damage will be done to the security and economy of the country Interestingly, the largest manufacturer of network security software is Checkpoint, headquartered in Israel.

The issue of network security is a strategic issue for countries; therefore, our country must be equipped with the latest network security technologies, and since these technologies cannot be purchased as software products, then the country's researchers must take this important and work in it.

Today, the Internet has become so accessible that anyone can access and use it, regardless of where they live, nationality, job or time. This ease of access exposes it to risks such as loss, abduction, distortion or misuse of the information contained therein. If the information was printed on paper and kept in a cage in the safe rooms of the relevant office, unauthorized persons would have to go through various fences to access it, but now a few hints of computer keys are enough for this purpose.

1.3.4 Network security history

The Internet was launched in 1990 as a network called Arpanet, which was affiliated with the US Department of Defense. The goal was to use interconnected computers to create a situation in which, even if major parts of the information system failed for any reason, the entire network could continue to operate, so that this information could be preserved. From the beginning, the idea of creating a network was to prevent the destructive effects of intelligence attacks. In 1990, a number of computers from universities and government centers were connected to the network, and researchers began exchanging information. With the emergence of unexpected events in the information, attention to the issue of security became more and more. In the year, Arpanet first encountered a global security incident on the network, later known as the "Morris worm." Robert Morris, a student in New York, wrote programs that could be accessed and reproduced on another computer, as well as infiltrated and reproduced geometrically. At that time, 3 computers were connected to this network. The program caused 10 percent of networked computers in the United States to crash in a short time. Following this incident, the Security Incident Foundation (IRST) was formed, which played an effective role in coordinating counter-security activities, training and equipping networks, and preventive measures. As the Internet became more common and popular, it became more and more secure. Among these incidents was the WINK / OILS WORM network security breach per year, Sniff 1989 per packet per year, the latter of which was published via e-mail, revealing information about users' passwords. Since then, information security attacks on networks & the internet have increased with each passing day. Although the Internet was initially developed for educational and research purposes, today it has found many commercial, medical, communication and personal applications that have made the need to increase its reliability even clearer.

1.4 Thesis Objectives

Security represents a major issue in internet literature. Internet security means providing solutions to combat abuse in this space. Internet security includes important parts and primary concepts related to the security of computer networks is issue of infiltration

into these networks. Therefore, researchers in this field are trying to offer a variety of strategies to combat infiltration. In the meantime, the use of machine learning tools play crucial role in detecting intrusion.

Accordingly, the objectives of this study can be expressed as follows:

- ✓ Investigate security challenges in Internet networks
- ✓ Investigating the significance of machine learning and classification in upholding security within Internet networks
- ✓ Providing a solution including both meta-heuristic algorithms and machine learning tools (deep learning with traditional learning), especially classification to detect attacks on Internet networks.
- ✓ Applying classification and feature extraction methods based on deep learning methods to detect intrusion

Combining meta-heuristic algorithms of GW[7] optimization algorithm and NSGAII[8], MOPSO[9] to optimize learning model parameters and feature selection.

Therefore, the main purpose of this research; this study proposes to employ metaheuristic-based feature selection methods for Convolutional Neural Networks (CNNs) models for intrusion detection problem. This approach essentially aims to enhance intrusion detection accuracy for providing better security of network.

1.5 Significance of the Tthesis

Today, the Internet has directly or indirectly affected many things. Although these networks have made many tasks easier and have significantly increased the speed of various tasks and also a lot of work is done on the Internet, but there are some problems

[7] Grey Wolf
[8] Non-dominated Sorting Genetic Algorithm II
[9] Multi-Objective Particle Swarm Optimization

with these networks. For example, a broken Internet can cause irreparable damage to various sections of human society, even for a few moments. Accordingly, maintaining the security and stability of these networks is very important and necessary. Given that hacking into Internet networks is main security problems, it is very important to deal with hacking into these networks. Given that intrusion and attacks follow a similar pattern, it is possible to provide a way to counter intrusion using machine learning tools and meta-optimization optimization algorithms. In the meantime, extensive study and study of anti-infiltration strategies is of great importance and necessity.

1.6 Thesis Structure

In this chapter, the generalities of the thesis were presented. The chapter aims to get acquainted with the basic concepts and objectives of this thesis. Following this thesis:

- In the second chapter, literature and work history will be examined.

- The third chapter is devoted to explain the research methodology.

- In the fourth chapter, we present the evaluation methodology for the proposed approach and provide an overview of the test results.

- The fifth chapter focuses on describing the overall results obtained from the proposed method.

2. LITERATURE REVIEW AND BACKGROUND STUDY

2.1 Introduction

The section will provide a brief overview of what is being pursued in this dissertation to provide an overview of what lies ahead. In the preceding section it was mentioned that one of most basic security problems of computer networks is the intrusion into networks with different purposes. Accordingly, it was stated what methods can be used to detect intrusion. This chapter tries to examine the research literature in detail and finally to review some studies related to the field of dissertation.

2.2 Attacks on Computer Networks

In this section, the types of attacks (intrusion) on Internet networks will be examined, one of which is the DDoS[10] attack (Patel et al. 2018). Table 2-1 categorizes the types of attacks on computer networks:

Table 2.1 Types of attacks on computer networks

Types of attacks	
Denial of Service (DoS[11]) & Distributed Denial of Service (DDoS)	
Spoofing	Back Door
Replay	Man in the Middle
Weak Keys	TCP/IP Hijacking
Password Guessing	Mathematical
Dictionary	Brute Force
Software Exploitation	Birthday
Viruses	Malicious Code
Trojan Horses	Virus Hoaxes
Worms	Logic Bombs
Auditing	Social Engineering
Spoofing	Back Door
	System Scanning

[10] Distributed Denial-of-Service
[11] disk operating system

2.2.1 DoS attacks

DoS attacks aim at disruption of the resources or services that users intend to access and use (disabling services). The most important goal of this type of attack is to deprive users of access to a specific resource. In this type of attack, attackers use a variety of methods to try to disrupt authorized users in order to access and use a particular service, and to disrupt the set of services that a network provides. Attempts to create fake traffic on the network, disruption of communication between two machines, obstruction of authorized users to access a service, disruption of services, are examples of other goals that attackers pursue. In some cases, DoS attacks play role as starting point that pave the road for the main attack. The proper and legal use of some resources may also lead to a DoS-type attack. For example, an attacker could use an anonymous FTP [12]site to access copies of illegal software, use disk storage, or create fake traffic on the network. It is important to note that engaging in any form of unauthorized attack or attempting to disable network protocols without proper approvals or permissions is illegal and unethical. (Chang et al. 2017).

The information provided here is solely for educational purposes and should not be used for any malicious intent. While there are various types of (DoS) attacks, its crucial to emphasize conducting such attacks is a violation of network security and can lead to serious consequences. DoS attacks aim to disrupt or disable network services, making them unavailable to legitimate users. These exploits can measure weaknesses in network protocols, overwhelming the targeted system with an excessive amount of traffic or exploiting specific weaknesses. Although there may be tools available on the internet that can be used to launch DoS attacks, their usage is highly discouraged and illegal in most jurisdictions. Network administrators should focus on safeguarding their networks, implementing proper security measures, and conducting legitimate testing and debugging procedures to ensure optimal network performance. (Laboratory 2000).

Smurf / smurfing: A type of DDoS attack where the perpetrator attempts to overwhelm the target server by flooding it with an excessive number of packets. By sending fake

[12] File Transfer Protocol

requests, computer networks expose the target device to a large amount of traffic to slow the system down to the point where it becomes inactive or vulnerable. This attack is generally a resolved vulnerability and is no longer common.

Fraggle attack: The Fraggle attack shares similarities with the Smurf attack, but the key distinction lies in the utilization of (UDP) instead of (ICMP). Like the Smurf attack, Fraggle attacks involve sending a large volume of UDP packets to broadcast addresses, causing congestion and overwhelming the targeted network.

Ping Flood: A Ping Flood attack, also referred to as ICMP flood, is form of (DoS) attack where the assailant overwhelms the victim's computer or network with a massive number of ICMP Echo Request (Ping) packets. By overwhelming the victim's system with these requests, the attacker aims to consume network resources, block services, or reduce the system's performance, potentially rendering it unresponsive.

SYN flood: SYN flood: In this type of attack, the benefits of TCP[13]-related three-way handshake are used. The source system sends a large set of SYN[14] requests without sending the final ACK[15]. This creates half-open TCP sessions. TCP stack causes the destination computer to overflow the connection buffer, disabling communication with trusted clients.

Land: A LAND (Local Area Network Denial) attack is a specific type of denial-of-service attack that entails sending a forged packet to a computer with the intention of disabling it. This security hole was first exploited in 1997 by those using the alias M3it.

Teardrop: Teardrop additional variant of (DoS) attack that capitalizes on a vulnerability present in TCP/IP stack implementation of specific operating systems. It targets the way fragmented packets are reassembled by the operating system. In a Teardrop attack, the assailant sends specially crafted fragmented packets to the targeted system. With

[13] Transmission Control Protocol
[14] synchronization
[15] acknowledgment

overlapping or overlapping offset values. When the targeted system tries to reassemble the original packets, the overlapping fragments cause the system's TCP/IP stack to become confused or crash. This occurs because the system fails to handle the overlapping fragments properly, leading to memory corruption or system instability. The impact of a Teardrop attack may vary based on the vulnerability and the particular operating system involved. In some cases, the target system may experience a crash or reboot, resulting in a denial of service for legitimate users.

Bonk: In this attack, the perpetrator sends corrupted or malformed (UDP) packets to the (DNS) on port 53. By sending these corrupted packets to the target machine's DNS port, the attack aims to overwhelm the system's resources, disrupt its performance, and potentially cause the system to crash. The corrupted packets exploit vulnerabilities or weaknesses in the way the affected Windows operating system handles incoming traffic on the DNS port.

Boink: This type of attack is similar to Bonk attacks. The difference is that instead of using port 53, multiple ports are targeted. Table 2-2 lists the most commonly used ports for DoS attacks.

Table 2.2 The most common ports used in DoS attacks

Service	Port	Service	Port
Echo	7	Portmap	111
Systat	11	SNMP	161/162
Netstat	15	HTTPS	443
Chargen	19	RADIUS	1812
FTP-Data	20	DNS	53
FTP	21	HTTP	80
SSH	22	POP3	110
Telnet	23	TACACS	49
SMTP	25		

2.2.2 DDoS attacks

Another DoS attack is a specific yet simple type of DoS attack known as Distributed DoS (DDoS) (Lin et al.2019). In this regard, several softwares can be used to perform this type of attack from within a network. Dissatisfied users or people with malice can disable services on the network without any influence from the outside world of their organization. In such attacks, attackers distribute special software called zombies. This type of software will allow attackers to take control of all or part of the infected computer system. After the initial damage to the target system, the attackers will carry out their final attack using a large set of hosts using the installed Zombie software (Zhang et al.2016).

The nature and approach of this attack share similarities with a typical DoS attack, but extent of damage inflicted by the attackers on compromised systems is influenced by the overall number of zombies under their control.

To protect the network, filters can be configured on external network routers to remove packets subject to DoS attacks. In such cases, another filter should be used that allows traffic to be viewed (source via the Internet) and an internal network address.

DDoS is fundamentally a synchronized assault on Internet services. In this method, DoS attacks are performed indirectly on the victim's computer through a large number of hacked computers. The services and resources that are attacked are called "primary victims" and the computers used in this attack are called "secondary victims". DDoS attacks are generally more effective than DoS attacks at disabling large corporate websites. (Jasim et al. 2022)

- Types of DDoS attacks: DDoS attacks are generally divided into three groups: Trinoo, TFN[16] / TFN2Kand Stecheldraht.

[16] Tribe Flood Network

- Trinoo:Trinoo is an attack tool that operates using a Master/Slave architecture. It coordinates multiple compromised systems to launch UDP flood attacks against a target computer. The process of establishing a Trinoo DDoS network typically involves the following steps:

Step 1: The attacker gains control over one or more compromised hosts and compiles a list of potential targets. This process is often automated through the compromised host, which aids in identifying other vulnerable systems to be exploited.

Step 2: Once the target list is prepared, scripts are executed to compromise these systems and transform them into either Master or Daemon nodes. Masters serve as the controlling hosts, whereas Daemons act as compromised hosts responsible for executing actual UDP flood attacks on victim's machine. It is possible for a single Master to control multiple Daemons.

Step 3: DDoS attack occurs when the commands send by attacker to the Master hosts. These Master hosts then instruct the associated Daemons to initiate a DOS attack targeting a specific IP address. Provided in the command. By coordinating a large number of individual DoS attacks from multiple compromised hosts, a DDoS attack is orchestrated, overwhelming the targeted system's resources and potentially causing it to become inaccessible.

-TFN (Tribe Flood Network) and TFN2K (Tribe Flood Network 2000): The TFN (Tribal Flood Network), similar to Trinoo, operates as a Master/Slave attack where a coordinated SYN flood is directed towards the victim's system. TFN daemons possess the capability to execute diverse attacks such as ICMP floods, SYN floods, Smurf attacks, making TFNs more intricate compared to Trinoo attacks.

TFN2K offers several advantages over the TFN. In TFN2K attacks, fake IP addresses are employed to carry out the assaults, which makes it more difficult to detect the source of the attack. TFN2K attacks represent something beyond simple outbursts;

meaning that they include attacks exploiting vulnerabilities of the system security for invalid and incomplete packages, thereby causing victim systems crash. TFN2K attackers no longer require logging into client machines to execute commands, unlike TFN masters, and can execute these commands remotely. TFN2K is very difficult to detect and more dangerous than many other attacks.

- Stacheldraht:The connection between the masters and attacker the in a Stacheldraht attack is encrypted; The agents have the capability to enhance their code and are capable of launching different types of attacks, including ICMP floods, UDP floods, and SYN floods.

2.3 Intrusion Detection Systems

Intrusion detection system is a system designed to monitor and analyze network traffic in order to detect suspicious activities. Detecting and reporting anomalies and suspicious activity is one of the main features of IDS systems.(Zhang et al.2016) Firewalls or firewalls are designed to prevent attackers from accessing resources, and an intrusion detection system reports and monitors activity, so these systems do not prevent intrusion but detect that intrusion has taken place or not.

This section describes the various components of intrusion detection systems.

- Activity: An activity is actually an action or event that is performed on a data source, it is attractive to the performer and it works for him. The activity can be both true and false, meaning that suspicious activity can be malicious and compromise the source, for example a type of "transfer control protocol" communication request that alternates from one IP address to another. The server side is established, can be an example of suspicious activity (Yousef El Mourabit et al.2015).

- Administrator:The Administrator is, in fact, the person responsible for drafting security statements for the organization. These individuals are responsible for making

decisions about how to implement and configure intrusion detection systems. The manager makes decisions based on alert levels, log history, and Session monitoring capabilities. They are also responsible for deciding how to respond to intrusion and ensuring that attacks are responded to effectively.

In many organizations, there is an Escalation chart, so security managers or network administrators are not at the top of this organizational chart in most cases, but tasks are always assigned to these people when implementing these types of security systems (Ibor et al.2018).

- Alert: An alert is a message that is sent through the analytics system to announce the occurrence of a suspicious event. This alert not only provides information about the type of activity performed, but also describes exactly what happened. For example, when a large number of ICMP protocol packets are entered into the server or a large number of unsuccessful login requests are generated, an alert is issued through the analyzer system. There is always a normal amount of traffic for a network. Warnings are issued when this traffic exceeds the set limit and exceeds the normal limit. We never want to be alerted when anyone pings from anywhere using the ping server command! But when this volume of ping packets exceeds the allowable limit on intrusion detection systems, we will definitely need to create and issue an alert (Purcell et al.201).

- Analyzer: An analyzer is a component or process that analyzes data obtained through a sensor. This process scans data to find suspicious activity. Analysts work by monitoring events and detecting whether a current event is suspicious, as well as following the rules that are activated in intrusion detection systems by the relevant processes (Hready et al 1990).

- Data source: The data source is raw information that intrusion detection systems use to detect suspicious activity. Data sources can include audit file, system logs, or network traffic where the intrusion detection system is located (Khraisat et al. 2019).

- Event:An event refers to an incident that transpires within a data source and signifies the presence of a potentially suspicious activity. An event may trigger an alert. Events are always logged for future reference. Events can report alerts for unusual events on the network. An intrusion detection system may start recording events when the email server's internal connection is locked. Events may indicate that someone is searching for and discovering information from your network. The event can eventually issue a warning that the volume of traffic passing through the network is out of the usual and defined range and needs to be investigated (Chang et al. 2017).

- Manager: A manager is a tool or process utilized by an operator to oversee an IDS. Any modifications to the settings of intrusion detection systems can only be made by communicating with the administrator.

- Notices: Notification is the process or method by which an administrator alerts an operator. Such notification may be done by highlighting or changing the color of a part of the management console or sending emails and text messages to the operator (Park et al. 201)].

- Sensor: A sensor is a crucial component of an IDS responsible for gathering and transmitting data to the analyzer. These sensors can be integrated into the network circuit either as a device driver or as a standalone black box, separate from the IDS. The sensor communicates its findings to the IDS, serving as the primary point of data collection.

- Operator: Operator is the person who is generally responsible for the intrusion detection system. This person can be the network boss or manager, a network user, or anyone else, but he or she should be responsible.

As described, the intrusion detection system is composed of various components and processes, all of which are put together to provide a true picture of the traffic transmitted in our network in real time. Figure 2-1 shows the various components of

this system together. It is important to note that data can be collected from various sources, all of this data must be thoroughly analyzed in order to be able to identify what is happening on network. An IDS is not primarily designed to block network traffic, however, certain types of IDS do possess the capability to perform this function. Intrusion detection systems are inherently monitoring, inspection and monitoring systems. (Breiman 200)].

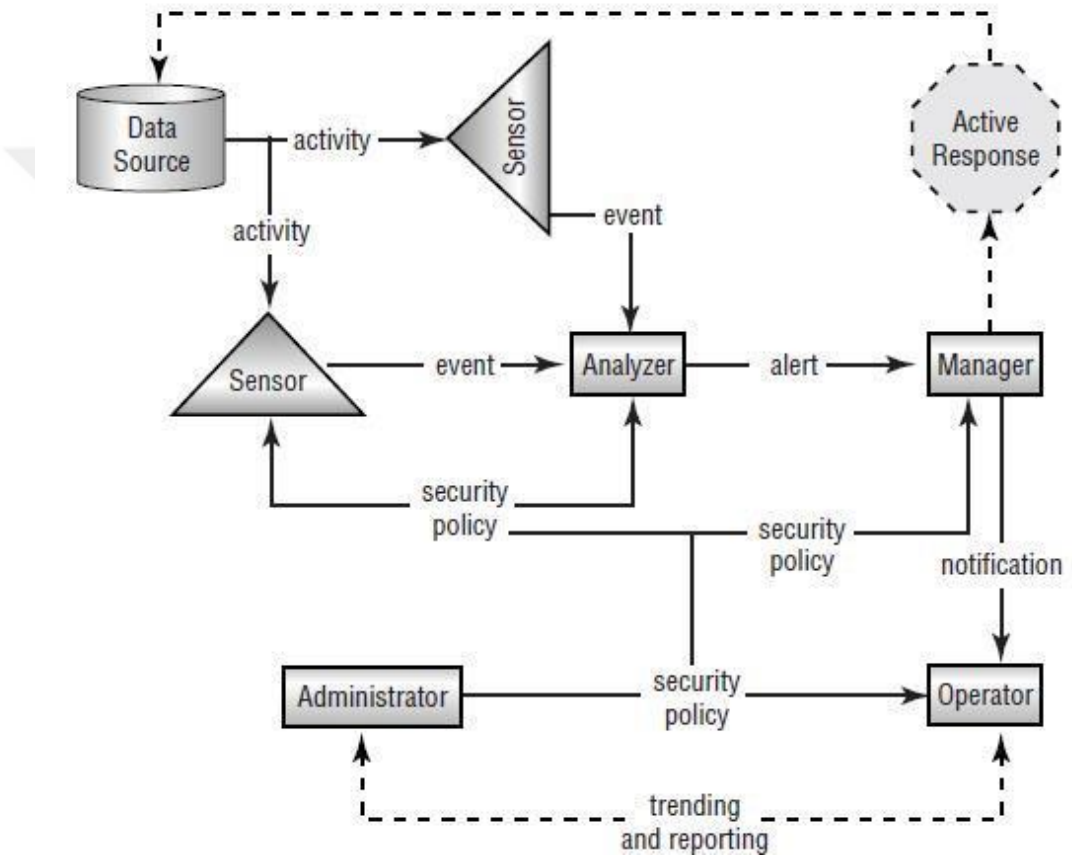


Figure 2.1 Components of IDS (Intrusion Detection System)

2.4 Methods of Detecting Cyber Aattacks

Most networks are connected via the Internet for sharing data and information. Although the sharing of valuable resources to increase operational efficiency has shaped the pattern of computer networks, it has created an integrated source for the easy spread of malware, thereby intensifying cyber-attacks in cyberspace.

Detection of cyber-attacks represents a major technique for attenuating the attacks in network. This includes responding to an unusual connection to report the detection of an attack pattern or profile within network. Intrusion detection is a crucial part for detecting cyber-attacks (Jasim et al. 2022).

Figure 2.2 shows the methods for detecting cyber attacks.

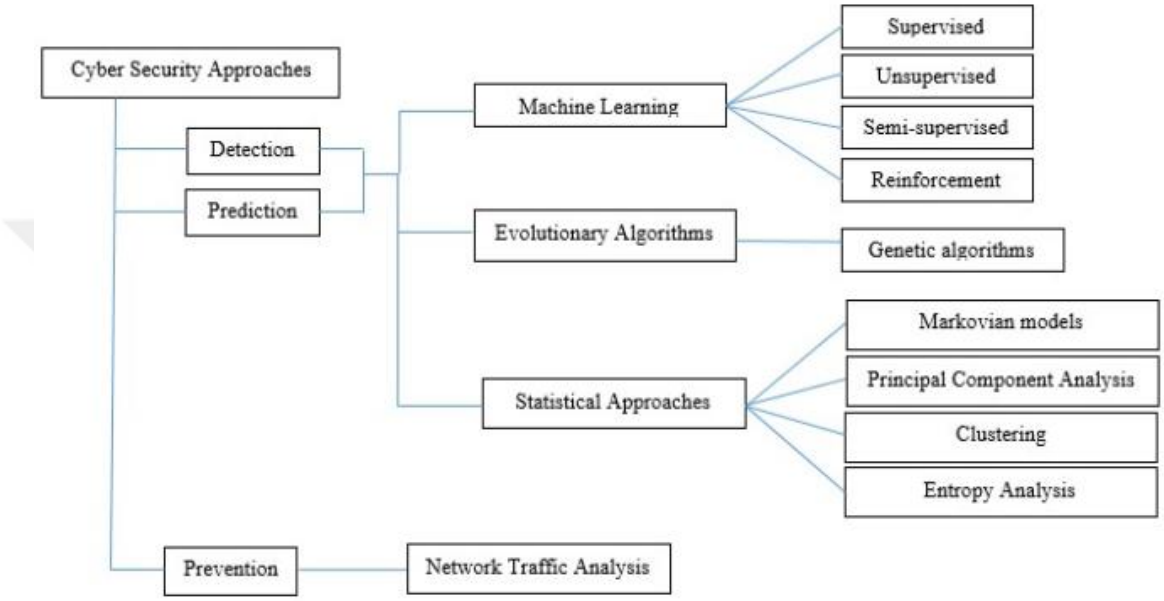


Figure 2.2 Methods of detecting cyber-attack

One of the most important ways to detect intrusion is to use machine learning tools. Figure 2.3 shows The Intrusion detection process on the utilization of Machine learning methods (Khraisat et al. 2019).

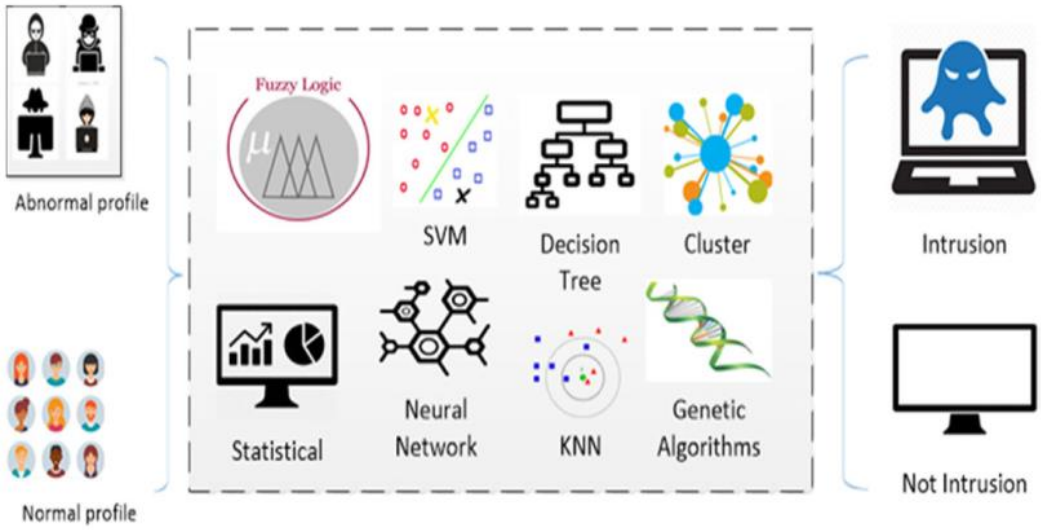


Figure 2.3 Learning-based attack detection approaches

2.5 Machine Learning and Classification

Absolutely, machine learning plays a significant role in identifying patterns and extracting knowledge from data. Machine learning is a specialized field of artificial intelligence (AI) that revolves around the creation of algorithms and models enabling computers to learn from data and make predictions or decisions without explicit programming. This science is very powerful in discovering knowledge due to the tools at its disposal. As the volume of data continues to grow and the limitations of machine learning tools become apparent, the field of data mining has emerged. Data mining builds upon machine learning principles but incorporates more advanced algorithms and tools specifically designed for handling large-scale datasets, commonly referred to as big data. Within the realms of data mining and machine learning, there exist a tools for classification, regression, and more. In the continuation of this section, the classification is described.

- Classification is considered a fundamental and critical function in machine learning to assign a new object or instance to one of the predefined categories. This practice has many applications, including detecting e-mail based on content or message header, distinguishing between cancerous and non-cancerous cells

by analyzing MRI scans. Another application involves classifying galaxies based on their distinct shapes.

- In classification, the input data consists of a collection of records, where each record is referred to as an instance or example. An instance is represented by an ordered pair (X, Y), where X denotes a set of attributes and y represents a special attribute known as the class tag or target attribute.
- The feature set used can be continuous or discrete. But the class label must be discrete. Continuity or discontinuity of the label feature is a very important and key component in distinguishing classification from regression. If the y label is continuous then the prediction operation will be regression.
- Classification is a practice whose purpose is to learn the objective function f, which maps each set of properties x to one of the predetermined classes y. The objective function commonly referred to as the classification model. Classification models serve various purposes and are particularly suitable for the following applications:
- descriptive models: The classification model serves as a descriptive tool that enables differentiation between objects belonging to distinct classes
- Predictive model: Classification models are often used to predict unseen record (sample) category tags. As shown in Figure 2-4, a classifier model behaves like a black box that automatically labels a set of new and unknown sample properties. (Khraisat et al. 2019)

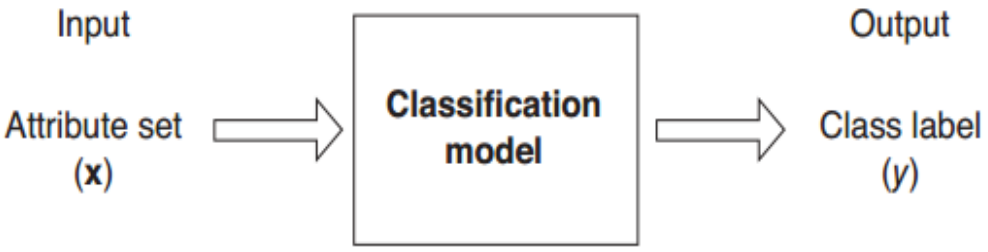


Figure 2.4 Input x tagging based on its feature set

Classification refers to a set of supervised learning tasks where both input features (X) and corresponding output labels (Y) are available, aiming to understand the connection between X and Y. In supervised learning, the primary objective involves grasping the association between the input and output, with an expert or observer providing the accurate output for each input.The dual of supervised learning, unsupervised learning. In this method, there is no monitor and only input data is available. In this method, the goal is to find order between the input data. In this case, there is a structure between input and output data that a particular pattern occurs more than other patterns and the goal is to Explore the occurrences and non-occurrences of events, a statistical concept known as density estimation.

Machine learning offers a variety of classification algorithms, such as support vector machines, decision trees, nearest neighbors, and neural networks. Each of these algorithms possesses unique strengths and weaknesses, highlighting the absence of overall superiority among them .

Types of classification algorithms in machine learning: Classification refers to learning how to assign a class label to an input sample. For example, classification can determine whether an email is spam or not. The class labels here are spam and non-spam, which must be converted to numeric values, that is, we set spam to zero and non-spam to one. Another example of classification is the classification of handwritten characters into existing characters.

Perhaps the problem of classification can be divided into four categories:

- Binary classification
- Multi-class classification
- Multi-label classification
- Unbalanced classification

In the following, we will briefly review each of these cases.

-Binary Classification:

Consider the scenario of distinguishing spam emails, which can be categorized into two labels: spam or n-spam, or in medical tests, it is determined whether a patient has a certain disease or not, so we have two labels of sick or non-sick. In fact, in binary classification, as you can see in the figure below, one class signifies the normal state, while the other class signifies the non-normal state. (Park et al. 2020).

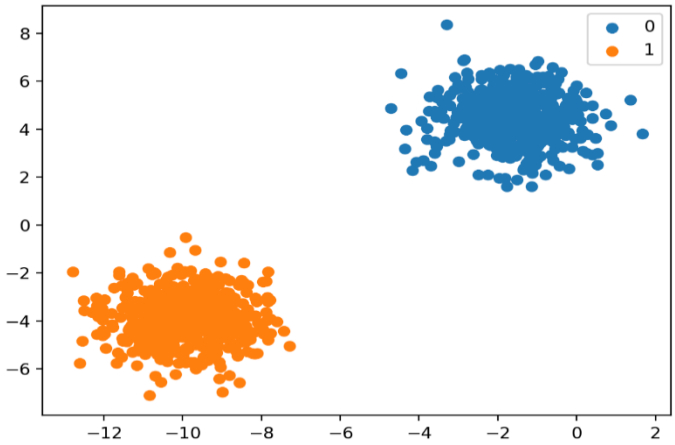


Figure 2.5 Binary Classification

Common algorithms for binary classification

- Logistic Regression
- K Nearest Neighbors
- Decision Trees
- SVM
- Naive Bayes

Multi-Class Classification: M-class classifications are group tasks that have more than two type of labels. For example, in the figure below we have three different classes. such as face classification, classification of plant species and identification of optical characters. Unlike binary classification, samples belong to a wide range of known classes. The number of class tags in some problems may be too many. For example, in a face recognition system, the model predicts whether a photo belongs to one of the many of faces in the system.

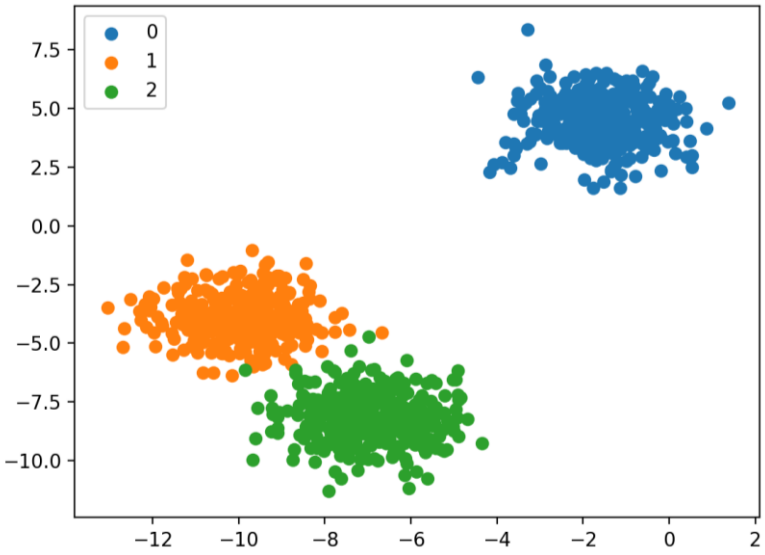


Figure 2.6 Multi-Class Classification

Some popular algorithms for classification problems are:

- K-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Random Forest
- Gradient Boosting

Multi-Label Classification: This represents a task where each sample can be assigned two or more class labels. For instance, in the context of photo classification, when a photo can contain several components in the image, a model can predict several labels in the photo, such as people, bicycles, apples, etc. In the figure below, the difference between two classification approaches is highlighted.

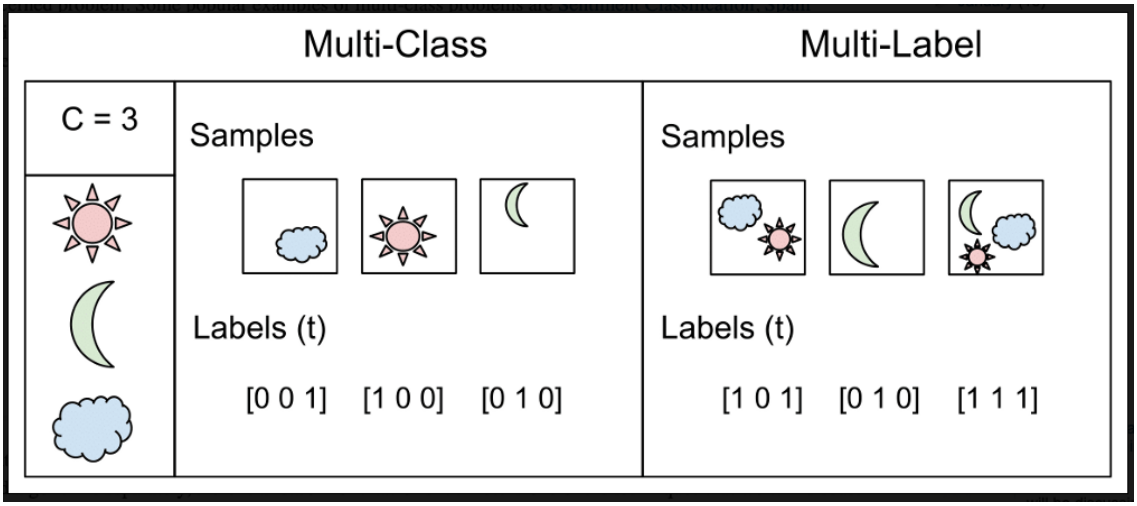


Figure 2.7 The difference between multi-class and multi-label classification

Binary and m-c classification algorithms cannot be applied straight to these problems, so versions of multi-label algorithms should be used. As:

- Multi-label Decision Trees
- Multi-label Random Forest
- Multi-label Gradient Boosting

Imbalanced Classification**:** In unbalanced classification, the training set is characterized by a significant majority of samples belonging to the normal class, while the non-normal class contains a relatively small number of samples. As you can see in the figure below, there are majority of dots in blue color and few dots in yellow color. Such as fraud detection, data outlier detection and medical diagnostic tests. For example, in cancer diagnosis tests, the large number of healthy people and a small number of cancer patients. (Miah et al.2019).

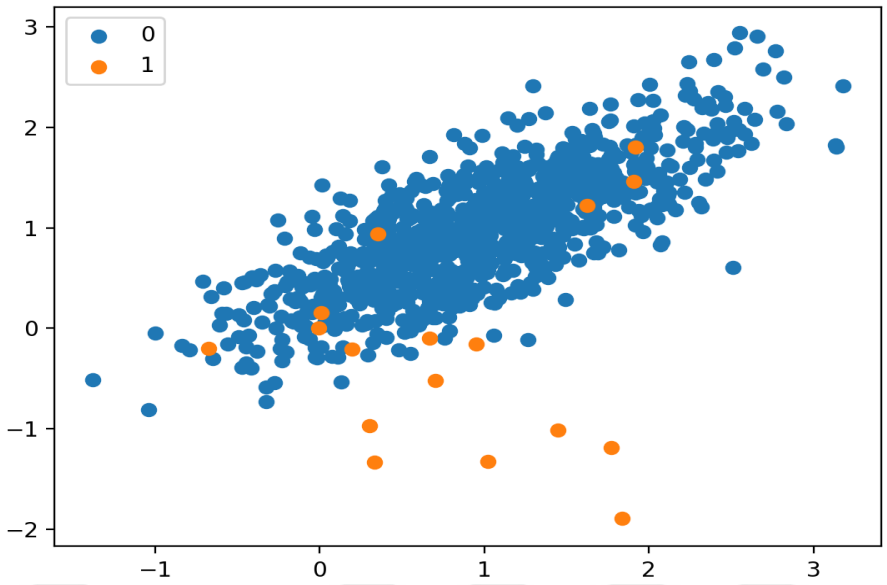


Figure 2.8 Imbalanced Classification

Special modeling algorithms called cost sensitive M-L algorithms can be used for unbalanced data. for example:

- Cost sensitive Logistic Regression
- Cost sensitive Decision trees
- Cost sensitive Support Vector Machine

We may also need alternative Metrics such as Precision, Recall, and F-Measure because reporting classifier precision may be misleading (Savita and Ayesha 2019).

2.6 Deep Learning

The inaugural use of deep learning dates back to 2006, marking the inception of this revolutionary concept. This concept was introduced as a new field of research in machine learning and was used in most cases for research related to pattern recognition, but now it is used in artificial intelligence-based practices. It has applications. In general, neural networks consist of a set of input-output units that are interconnected, so that each connection between units has a weight. The utilization of in-depth learning enables the application of computational models comprising numerous layers of

processing to acquire the ability to represent data with multiple levels of abstraction. Network architecture in deep learning is usually considered as multi-layered to compute more abstract features as nonlinear functions than low-level features. In short, the concept of deep learning is the learning of features of a high level of data that this learning takes place nonlinearly and in successive layers in the neural network. As the number of network layers increases in deep neural networks, the learning of features improves, naturally, with an increase in the number of network layers, a substantial amount of data becomes necessary for training the network. As hardware capabilities continue to advance, including the growth of memory capacity and image sensor resolution, increasing computing power as well as reducing power consumption in new hardware, the use of deep learning compared to traditional computer vision methods has numerous advantages, including this advantage we can point to the increase in performance and usefulness. Deep learning models are trained instead of being planned, so they need less adjustment and analysis, and due to their greater flexibility, they use data in today's world in large volumes. The deep learning method can be placed in different categories according to its application. The most important types of deep learning models are convolutional neural networks, limited Boltzmann machines, generative conflict networks and automatic encoders, which are used in many users of machine vision, bioinformatics and natural language processing (Savita and Ayesha 2019).

DNN[17] is indeed one of the most prevalent architectures in deep learning. Typically, a deep neural network (DNN) consists of an input layer, multiple hidden layers, and an output layer. The input layer receives the input data, and as it passes through the hidden layers, the network undergoes a sequence of transformations and computations. This process enables the network to learn and extract pertinent features and representations from the input data. Finally, the output layer generates predictions or estimations based on the learned features, often represented as mappings between inputs and outputs.

[17] Deep Neural Network

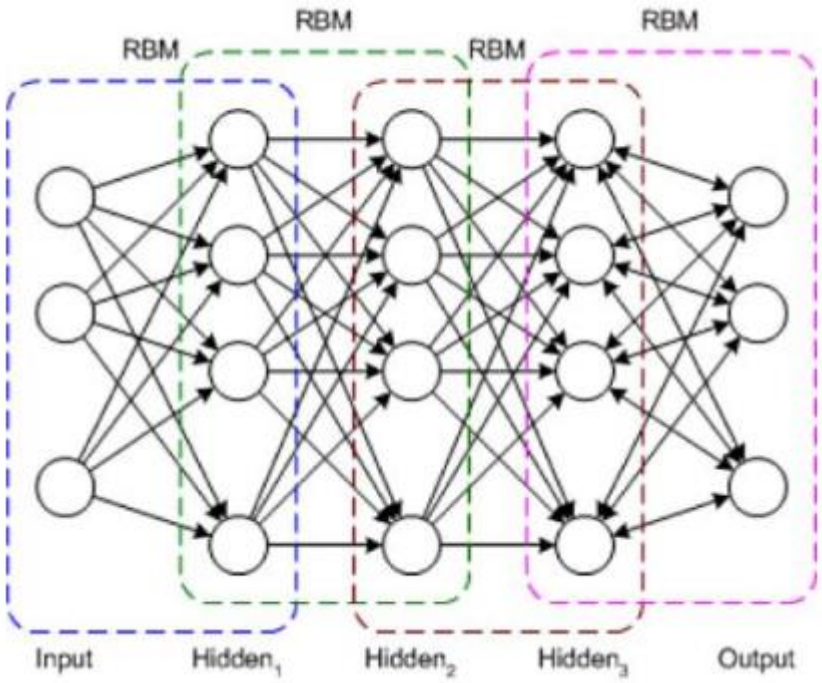


Figure 2.9 NNS with two hidden layers

We can observe the representation of a NN comprising 2 hidden sheet, showcasing its structure. Each node in the model corresponds to a neuron that receives the output from the preceding sheet, including the bias of a particular neuron. It then calculates the weighted average of its inputs, which refers to the total input. For each hidden unit j, yj produces a numeric output, bj and wji the elements of the weight matrix are one layer. The activation function, denoted as f(), is responsible for determining the achievement of a conceal unit in a neural network.

$$y_j = f\left(\sum_{n=1}^{\infty} w_{ji} x_i + b_j\right) \tag{2.1}$$

In general, NN consists of three main components: an input, one or more hidden layers, and an output, and an output layer. In feed-forward NN, details moves from the input sheet, reach through the hidden sheet, and ultimately reaches the output sheet. The input layer is responsible for receiving the initial data values. These values are then multiplied by their respective weights and forwarded as inputs to the subsequent layer, referred to as the hidden layer(s). The hidden layers perform computations on these inputs using activation functions, generating intermediate representations or features. The quantity of

hidden sheet can be customized to suit problem. The structure feed neural networks connected. Each layer is made up of several neurons, and one neuron in one layer is joined to every neuron in the following layer. The deep neural network uses a descending gradient optimization algorithm and the backward propagation method to new weight on neurons, which is rested the calculation of the mistake function. Occurs to generate output values and then output values to calculate the difference in the middle of values foretell by the mesh and the values collect; that is, the delta value is redistributed through the network. For each weight, the amount of delta output and input activator are multiplied to obtain a gradient, and finally a percentage of the gradient is used to adjust the weight. Each hidden layer unit receives as input the collect the weighted prize of the preceding similitude units and then a trigger function request it and determines the value of each unit as the output. Compares the data obtained at the network output that determines the data class with their actual value and changes the weight of the edges in akin an action the square of the difference between the guessed at value and the existing value is reduce. This weight change occurs from the output to the input. In the following, we describe some modern deep learning architectures that have been considered by many researchers due to their high accuracy in machine vision.

CNN[18] is a deep learning algorithm specifically designed for processing and analyzing images. It leverages the concept of approval tenable density and preference to different aspects or equipment within an input figure, symbol it to categorize and differentiate among them. One of benefits of Convolutional Neural Networks (CNNs) is that they require minimal pre-processing compared to other classification algorithms. While traditional methods often rely on manually engineered filters, CNNs troop capability to receive and acclimate these filters or appearance through extensive training. The architecture of a ConvNet is activated by organization & connection patterns a neurons in the humanitarian, particularly a visual cortex. Each neuron in a ConvNet responds to a specific district of the input image admitted its " most overfilled." By using a set of overlapping receptive fields, the ConvNet is able to cover the entire visual area and extract meaningful features. This arrangement allows the internet to capture stratified portrayal of the aid data, starting from low status appearance like perimeter & textures

[18] Convolutional Neural Network

& progressing to more complex and abstract features. Away absorbing convolutional sheet, league layers, and totally connected sheets, CNNs can effectively capture spatial relationships and patterns present in images. These networks Significant accomplishments have been achieved in a wide range of computer vision tasks, including but not limited to image classification, object detection, and image segmentation, due to their ability to automatically learn relevant features from the data. In summary, CNNs provide a powerful framework for processing and understanding images by mimicking the behavior of neurons in the visual cortex. Their unique architecture and feature learning capabilities make them well-suited for a wide range of computer vision applications (Park et al.2018).

In image classification, CNN takes an image as input, analyzes it and classifies it into certain groups, for example, dog, cat, tiger, lion, etc. But the important thing is that the computer does not see the image as we do. The computer sees an array in the form of h×w×d, where h is the breadth of role, w is the width of the image, and d is the dimensions of the image.

It also has a photo with an array of 1x6x6 length and width of 6 pixels and 1 color channel; This means that the photo is black and white, not color. Each of the pixels has a value between zero and 255, which actually shows the intensity of each pixel; For example, in a black and white photo, the number zero represents black and the number 255 represents white; The closer the numbers are to zero, the darker they are and vice versa (Jasim et al. 2022).

Structure of CNN

A CNN entail of two general parts:

- Feature Extraction
- Classification

In fact, when a photo is entered into a CNN network, it first enters the feature extraction stage. During this stage, Convolution layers are applied to the input photographs in

multiple iterations, the ReLU activation function. Then incoming photos are entered into the classification; In this stage, flattening is done first, Subsequently, the outputs are fed into a for multi-class classification tasks, a fully connected layer is employed, and subsequently, a Soft ax function is utilized. In the case of binary classification, a sigmoid function is applied. So that the photos are based on Possible values are classified between zero and one.

First step: feature extraction (Feature Extraction); Convolution layer

This layer is the first step in extracting the input image features. In convolution layer, network learns the features of the photo by using filters (filter/kernel). As we said, the computer sees each input image as a matrix of pixels, each of these pixels having a value between 0 and 255; For example, consider a 5x5 pixel image, where the value of each pixel is zero or one instead of 0 to 255 for convenience and better understanding.

- Stride step:The filter's stride represents the number of pixels it moves as it traverses the image.moves on the input matrix. When we consider the Stride value to be 1, it means the filter moves 1 pixel each time, and if we consider 2, it means the filter moves 2 pixels each time.

- Padding: In the Padding technique, add zero to the input image matrix, which is called Zero-Padding.

- ReLU: In the feature extraction stage, an activation function (usually ReLU function) is applied after each convolution layer. ReLU is an activation function for the output no linearization operation.

- Pooling layer: In this step, the goal is to reduce the dimensions of the input image and keep only the important information or pixels and remove the rest. When the input image is very large, we can reduce its size to a great extent. This work is also called Down sampling or Subsampling.

Various forms of pooling exist, encompassing:

- Maximum Pooling
- Mean Pooling
- Aggregate Pooling

In most cases, Maximum Pooling is used in Convolutional Neural Network (CNN). In Maximum Pooling, the pixel that has a larger value compared to the remaining one, in Aggregate Pooling, the average value is selected, and in Mean Pooling, the sum of all values in each Feature Map is selected (Savita and Ayesha 2019).

The second phase of Convolutional Neural Network (CNN):

-Classification: As we mentioned before, in this Segment of the CNN , flattening is done first. This means that the output matrix from the first stage (feature extraction) must be converted into a vector; That is, if our matrix is 30x30, it will be converted into a vector or array of 900.

This vector is then imported into a quit associated sheet. A fully connected sheet is actually a multi-sheet perceptron (MLP / Multi Layer Perceptron) in which all the nodes of the prior sheet are connected to all the nodes of the next layer. The purpose of using the Fully Connected layer is to classify the input photos into different classes according to the features we have received from the first step. Finally, a softmax purpose is bid to it to determine the probability that the input image belongs to any class (eg cat, dog, etc.) in the output.

2.7 Feature Selection

Feature selection has been considered in many fields of research and applications, such as text processing, Internet documentation, and analysis of gene state arrays whose datasets are more than ten, hundreds, or thousands of variables(Lin et al. 2009). Feature selection goals include the following three:

- Improving the performance
- Create faster and more efficient estimators
- Provide a better understanding of the processes used to generate data (Hu et al. 2021).

It seems useful to use all the emphasize in the feature set, including the unrelated features, to estimate the relationship between input and output. In practice, however, there may be two major problems with the use of irrelevant features:

Redundant and irrelevant features increase the cost of calculations.

Duplicate and irrelevant features may over fitting.

Irrelevant attributes do not provide useful information, and duplicate attributes do not provide more information than currently selected attributes. In the field of inductive learning with supervision, feature selection selects a set of candidate features based on one of the following three methods:

- A certain size of the feature subset that optimizes an evaluation criterion.

- The smallest subset of attributes that satisfies a specific condition on the evaluation criteria.

- In general, select a subset of features that best fit between size and evaluation criteria.

Proper use of a feature selection algorithm improves inductive learning both in terms of generalization capacity and learning speed or reduces the complexity of the inductive model (Nadiammai and Hemalatha 2014).

If too many features are used to analyze the data, the method may be cursed later. Since not all pre-extracted properties contain useful information, the purpose of selecting a property is to choose properties that have a lofty separation strength for the data set.

In general, there are two basic ways to choose a feature. The first method is individual evaluation (evaluation of a feature), and the second method is the evaluation of a subsection of identify. identify rating is known as individual rating. In a single evaluation, the weight of each attribute is determined by the degree of relevance and appropriateness. In evaluating a subset of attributes, a subset of the characteristics of the selected candidate uses the search method.

The feature selection process typically involves four main steps, which are embellish in figure 2.10:

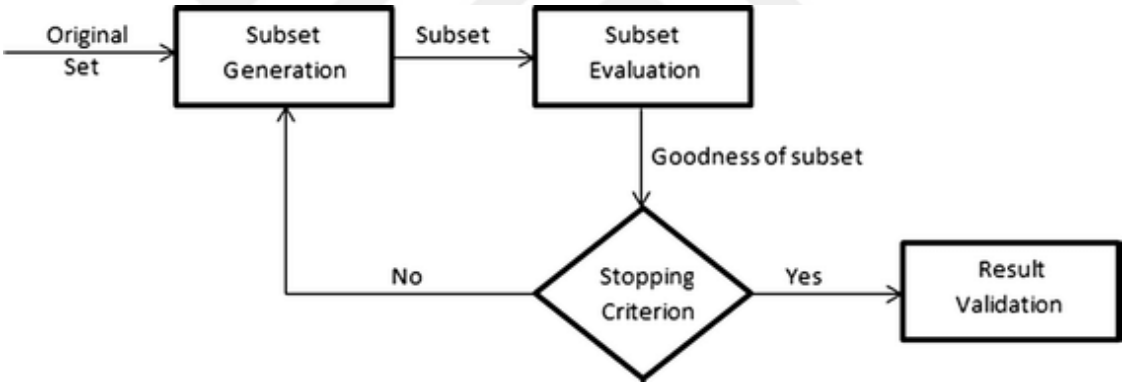


Figure 2.10 Four basic steps in feature selection

There are two general ways to select a feature:

-Filter Method
-Wrapper Method

The Filter Method acts as a preprocessing method, ranking the properties and selecting the properties that have the highest order for use in the estimator. In the wrapping manner, the criterion for selecting a feature is the coherence of the method. That is, the manner is embedded in a search algorithm that aims to discovere the best proper of features that provide the highest estimator performance (Zhou et al. 2020).

2.7.1 Filter method

The Filter manner avail the variables ranking techniques as the main criterion for selecting the attribute by sorting method. Ranking methods are used based on easy performance and success reported in practical applications. An appropriate ranking criterion uses scoring for variables and a threshold value, and the score of any variable (attribute) that is lower than the threshold is removed. Ranking methods are considered as Filter methods, because they are used to Filter (delete) variables that are not very suitable before the machine learning operation. Shape 2.11 demonstration the general steps of the filter manner:

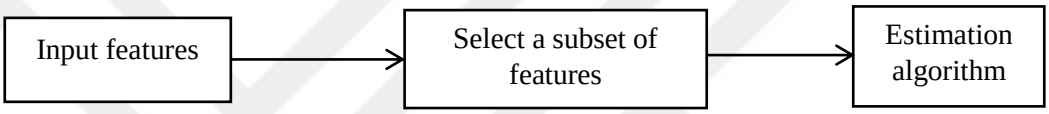


Figure 2.11 General process of feature selection by filter method

2.7.2 Wrapper method

The wrapper method uses the estimator as the black box and the estimator efficiency as the objective function to evaluate the feature subset. Since 2^{M} subset evaluation is Np-hard problems, relatively optimal subset of features can be selected using a search algorithm that selects subset revelations. The envelope method in selecting the feature is divided into two general categories which are:

- Sequential selection algorithm
- Ultra-innovative search algorithm

The sequential selection algorithm starts with an empty (full) set and achieves the vastest extent for the objective function by adding (removing) the attributes to (the) set. In order to rate the option turnover, conditions are created in which the avail of the target function is incremented step by step, and this act hang on until it reaches the

maximum value of the objective function with the least number of attributes(Zhou et al. 2020). The ggeneric act of the Wrapper manner is cluen in shape 2.12

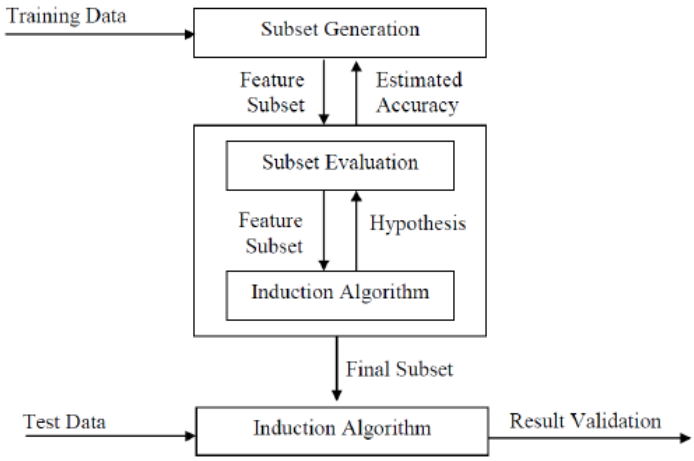


Figure 2.12 General process of feature selection by wrapper method

As mentioned, feature selection is widely used for machine learning tasks, examples of which are: feature selection for categorization, feature selection for regression, feature selection for clustering.

2.8 Meta-heuristic Algorithms

Heuristic and meta-heuristic methods are used in hybrid optimization. Although the goal of heuristic methods is to obtain ideal solutions, it does not guarantee that an optimal solution will be achieved (Doreco et al. 2006). Meteorological methods are used to obtain optimal answers. Optimization algorithms can be classified into two general divisions including precise and proximate algorithms. Approximate algorithms can achieve good (near-optimal) answers in a shorter time. They are categorized into three classes: heuristic, meta- heuristic and hyper heuristic algorithms. The major challenges of heuristic algorithms is early convergence of local optimal points; the problem addressed by innovative algorithms. Meta-heuristic algorithms are applicable in numerous problems. Although several meta-heuristic algorithms have been developed, there is lack of definition of the term Meta heuristic or meta-innovative. However, meta-heuristic methods represents various properties that make them ideal

solution in various fields. Unlike heuristic methods, meta-heuristic algorithms are not problem-dependent; In essence, meta-heuristic algorithms offer versatile outs to a broad spectrum of optimization obstacles. These algorithms employ advanced probe materiel that utilize experiences and information gathered during the quest flow, often in dole of a memory mechanism. the enables them to effectively navigate and explore the solution space, focusing on promising regions. Meta-heuristic methods differ from traditional optimization algorithms by providing more general and flexible search strategies. They are particularly adept at avoiding the pitfalls of local optimization. One crucial aspect of these algorithms is striking a dynamic balance between diversification and intensification strategies. Diversification involves conducting an extensive search across the solution space, while intensification focuses on leveraging the acquired knowledge to concentrate the search on more favorable areas. By dynamically balancing these strategies, meta-heuristic algorithms straightforward the hunt towards areas of the solution space that hold better solutions. Simultaneously, they avoid investing excessive time in areas that have already been explored or contain inferior solutions. This adaptive approach optimizes the turnover and effectiveness of the hunt flow. In summary, meta-heuristic algorithms are advanced and flexible search techniques that offer powerful solutions for escaping local optimization. They leverage the interplay between diversification and intensification strategies to impressively prospect the out area and identify improved areas.

Owing to the inability of accurate solution methods to solve all-out optimization knots, the utilization of meta-heuristic algorithms necessary due to their ability to generate optimal or semi-optimal solutions. Metaphorical algorithms fall into several important categories, including nature-inspired and non-nature-inspired. The following are some of these algorithms. These algorithms include dummy habitation algorithm, bat algorithm, biogeography optimization algorithm, dove optimization algorithm FA, differential evolution algorithm, GOA , GA, GWO, PSO article He mentioned many other algorithms.

Figure 2.13 provides a general classification of optimization algorithms.

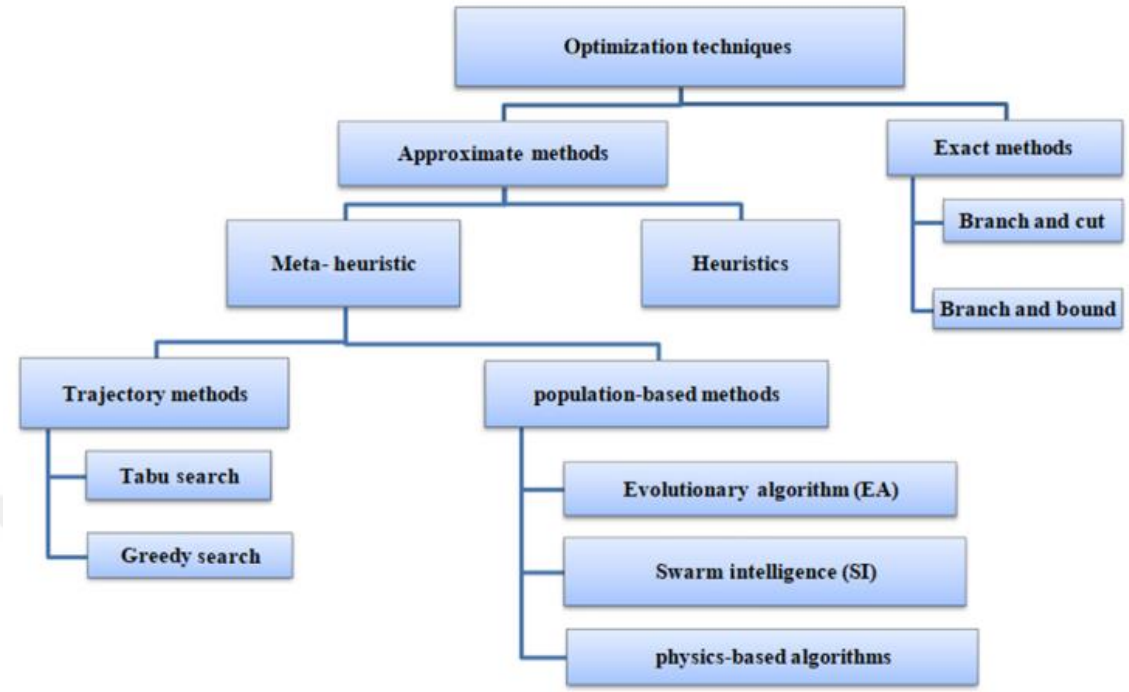


Figure 2.13 General classification of optimization algorithms

In this section, we tried to introduce the meta-heuristic and evolutionary algorithms, in the next section, a complete explanation of these algorithms is provided. Also, some examples of algorithms will be examined and their applications will be explained.

The origin and origin of meta-heuristic algorithms is well-found Darwin's notion of progress of inherent option (1859). According to this view, only suitable and strong people can survive in war and strife for survival in nature and reproduce to build the next generation.

2.8.1 General process of meta-heuristic algorithms

In general, it can be said that the most important parts of meta-heuristic algorithms include the following steps:

- Representation of the environment

- Evaluation function
- Population (collection of answers)
- The process of selecting parents
- Variation operators
- The process of selecting the living (selecting better people to build the next generation)
- Stop condition

As shown in Figure 2.14, meta-heuristic algorithms operate on the basis of populations of answers. Initially, a set of answers needs to be generated for use in the meta-heuristic algorithm. This answer must be defined for the meta-heuristic algorithm. In many cases, directly defining the problem in the context of the main problem cannot be accomplished within the meta-heuristic algorithm. Instead, a mapping process is typically required to transfer the problem-solving space to the exploration realm of the meta-heuristic algorithm. The solution is therefore determined by the data structure that indirectly describes the space of the main problem. In this case, there is a similarity between the Meta heuristic algorithm and evolution in nature, in which the genotype is mapped to the phenotype, and vice versa. Mapping phenotype to genotype shows how the representation of the gene space reflects the characteristics of the real space. In other words, the properties of real space represent the solution to the scope and context of the main problem. The initial population must be generated before operations on the Meta heuristic algorithm can begin. The initial population generation in Meta heuristic algorithms must be random. The basis of meta-heuristic algorithms is a meta-heuristic search in which a group of community-centered are selected to generate a generation with operators such as Cross over and Mutation. After the production of the survive selection is completed, based on the fit between the old people and the recently produced people, the live selection operation is performed and the more suitable people are transferred to the next generation. Fitness is calculated based on a function, often called the fit function. In meta-heuristic algorithms, the algorithm tries to optimize and optimize the generated answers in each iteration, and this process continues until the meta-heuristic algorithm achieves the best answer or the maximum number of iterations allowed (Uysal et al. 2019).

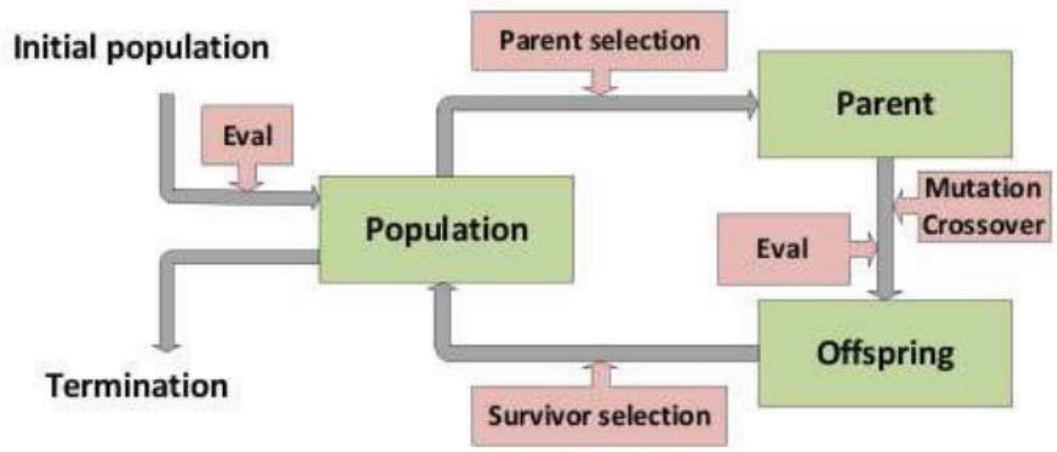


Figure 2.14 Cycle of MA

2.8.2 Steps of MA

The search steps of meta-heuristic algorithms are divided into two parts :

- Exploitation stage
- Exploitation stage

In the Exploitation stage, new solutions are tried to be identified and identified, but in the Exploitation stage, a search is made in the current solution space and, if possible, the obtained answer is optimized. The two cases mentioned in the search for meta-innovation are intertwined. High population variability increases the algorithm's detectability, and if population variability is lost, the meta-heuristic algorithm will be subject to Premature Convergence (Dincalp et al. 2018).

Exploitation and Exploitation rates in meta-heuristic algorithms can be adjusted by control parameters. For example, in a genetic algorithm, this is regulated by parameters such as population size, type of intersection, and mutation rate. In order to properly adjust these parameters, these parameters can be placed in the people in the population and this problem can be solved along with solving the main problem.

Exploitation and Exploitation rates in meta-heuristic algorithms can be adjusted by control parameters. For example, in a genetic algorithm, this is regulated by parameters such as population size, type of intersection, and mutation rate. In order to properly adjust these parameters, these parameters can be placed in the people in the population and this problem can be solved along with solving the main problem.

2.8.3 Classification of meta-heuristic algorithms

Different aspects must be considered in classification of MA. One aspect of these classifications is grounded in the concept from which these algorithms are modeled. Algorithms that are inspired by natural behaviors are divided into the following categories based on the classification made in:

- Stochastic algorithms
- Evolutionary algorithms
- Algorithms related to physical processes
- Probabilistic algorithms
- Swarm algorithms
- Immune algorithms

Among these, Evolutionary algorithms, physical, probabilistic, swarm and immune processes are among the meta-heuristic algorithms and have many applications in optimization problems (Brownlee 2011).

2.9 Literature Review

Proposed model by Farnaz and Jabbar leverages the random forest classifier along with various preprocessing and analysis techniques to achieve accurate classification results. The accuracy and correlation coefficient provide insights into the model's effectiveness in predicting and classifying the target variable. (Dwivedi et al. 2020).

Chang and colleagues proposed a breach prevention method informed by MA. Through this approach, the random forest heuristic is utilized to pick the features and the SVM[19] algorithm is used for classification. The Support vector machines algorithm has been used once by selecting 14 features and once by selecting 41 features for evaluation (Chang et al. 2017).

Manairhoe and Ahmed have researched the application and assessment of MA models in network-centric intrusion detection systems. 4 Random forest algorithms, decision stamp, random reduction gradient and Naïve Base are used. Various attribute selection techniques were used. The NSL-KDD99 collection of data was used for testing and assessmentThe findings indicated random forest algorithm has better detection accuracy and error than other algorithms. Also in this method, the the RF algorithm required a training time of 10.23 seconds. (Agrawal et al. 2021).

Patel et al. Proposed re-labeling of unlabeled labeled data employing a blend of supervised learning methodologies. The suggested technique comprises four sequential stages: 1- Selecting important features and deleting redundant features. 2. Clustering and data set training. 3. Build a learning model with supervision and using group classification. 4. Test specific parts of data. Our results show that the accuracy value is 81.29% contrasted against the initial labels. In addition, the group approach introduced utilizing LogitBoost and RF and the original labeling is 90.33% accurate and with new labeling for both educational and experimental data. The UNSW-NB15 dataset has been used for testing and evaluation (Maniriho and Ahmad 2018).

Park et al. reviewed the evaluation results and detection efficiency of different types of attacks using experimental experiments. They used the random forest algorithm and the Kyoto 2006 + data set (Park et al. 2018).

Singh and Venkatesan have used a combined approach of two machine learning algorithms called random forest and K-Means. They also utilized a hybrid approach

[19] Support vector machines

incorporating both signature-based and anomaly-based intrusion detection techniques. Using the advantage of two intrusion detection methods increases the efficiency and accuracy of the intrusion detection system. This paper focuses on a combination of Gaussian mixed clustering approaches with stochastic forest classifiers and K-Means clustering for intrusion detection (Singh and Venkatesan 2018).

Kumar et al. carried out research to establish an intrusion prevention system built upon artificial intelligence principles, focusing on outlier detection. By evaluating multiple algorithms and employing various evaluation metrics and techniques, they identified effective models that demonstrated strong performance in detecting intrusions.(Kumar et al. 2020).

Deborah et al. showcased the examination of network attack detection systems employing diverse machine learning methodologies. The proposed method utilizes ML algorithms such as REP Tree and Naïve Bayes J48 and random forest, and also uses the NSL-KDD99 data set for testing. The outcomes indicate the random forest algorithm has a better detection rate in comparison to other algorithms (Deborah et al. 2019).

Rani et al. showcased the examination in which feature selection is performed using a random forest algorithm in network data. Features are classified and analyzed by various machine learning algorithms such as: NB, KNN, LR, and DT. The proposed method consists of several parts: The data entry is entered into the system. The data is preprocessed. Pre-processed data is divided into experimental and training, with 75% for training and 25% for testing. In the next step, the classification algorithms are implemented (Rani et al. 2019).

Hsu et al. used group stack learning derived from network IDS, which is a fusion of self-cryptographic models, Support vector machines and random forest. The NSL-KDD & NB15 data sets used for testing and real records, which are about 300 million daily records. The proposed method includes modules for receiving NTD data pre-processing, and anomaly identification (Hsu et al. 2019).

Yihunie et al. proposed an approach that strives to discover an efficient classifier capable of effectively handling anomalous traffic in the NSL-KDD dataset with exceptional accuracy and minimal errors. Five distinct machine learning algorithms were employed within this approach. The outcomes reveal that the RF algorithm exhibits superior performance when compared to the other four algorithms. The NSL-KDD dataset was utilized for testing purposes. (Yihunie et al. 2019).

TaoLu et al. suggested a model that integrates the sampling technique of SMOTE & ENN were applied in conjunction with a random classifier algorithm. The NSL-KDD data set contains irrelevant and redundant data, which increases the complexity of time and resource consumption and reduces the detection rate. The results show that the combination of sampling method from the less hybrid part and the edited law of the nearest neighbor can cause higher accuracy in the intrusion detection system (TaoLu et al. 2019).

Miah et al. suggested an innovative approach to improve ID action rate using clustering and RF sampling. The suggested methodology first classifies the data whether network traffic is offensive or not, if network traffic is offensive it classifies it and then subdivides the attack (sub-attack) using clustering and sampling to deal with Class imbalance and random forest have been employed for classification purposes(Miah et al. 2019).

Numerous IDS introduced in the past few years. These systems rely on ML algorithms such RF & SVM, etc. Each algorithm has advantages and disadvantages. An algorithm works very well in detecting some attacks and performs poorly in some attacks. Shan et al. Have developed a consistent group system. In the proposed method, they take the advantages of each algorithmic and try to improve the intrusion detection by creating a group method (Gao ET AL. 2019).

Sah and Banerjee have studied various types of IDSs and feature reduction methods and classification techniques such as KNN[20], SVM, RF[21], NB[22]. The primary objective of thi article is to offer an improved approach. Feature selection increases accuracy. There are two feature reduction methods called PCA, REF. PCA extracts a subset of initial dataset. REF is utilized to select some data set features. The main task of REF is to reduce and eliminate irrelevant features by the recursive method. In this paper, REF is used to reduce the features due to the low build time of the models (Kaja ET AL. 2019).

Sallam et al. One of the attacks on computer networks is the DDOS attack,have proposed a method in which a database is divided into a set of groups using a classification of different types of attacks. Decision Tree, Random Forest, KNN, NB, AdaBoost are used for classification. It is a framework with various preprocessing and actual programming for DDOS detection in internet through the application of machine learning techniques. The proposed method consists of 6 steps. Pre-processing, data wiping, fashion selection, data training and testing, feature selection and classification. Random forest utilized to features selection and DT, NB, KNN, AdaBoost algorithms utilized for classification. Two modes are utilized for classification. In the first case, it only examines the features of the DDOS attack, and in the second case, instead of DDOS attack, the feature selection utilized entire data set. In the first mode, AdaBoost is utilized, which is 97.34% better than other algorithms, and in the second case, the decision tree is 94.68% better than other algorithms (Sallam ET AL. 2020).

Zhou et al.addressed the growing demand for healthcare services and the resulting increase in medical waste generation, which has begun to exceed the capacity for proper management. The research presents a DL methodology for the identification and classification of medical waste, leveraging the advantages of deep NN in show classifications. While deep learning is a powerful technique, its reliance on large amounts of data can limit its practicality. To address this issue, the authors propose a DL-based classification method that utilizes the ResNeXt architecture, a suitable DNN for real-world implementation. Transfer learning techniques are also employed to

[20] K-Nearest Neighbor
[21] Random forest
[22] NB

enhance the classification results.The focus of the study lies specifically on the urgent need for accurate medical waste classification within the context of environmental protection. The suggested approach is implemented to a data set of 3480 images, resulting in successful identification and classification of eight different types of medical waste with an impressive accuracy of 97.2%. The average F1-score, calculated using five-fold cross-validation, also achieved a high value of 97.2%. This research makes a contribution by DL-based solution for the automatic detection and classification of various medical waste types, demonstrating a high level of accuracy and precision. The authors believe that this approach harnesses the power of artificial intelligence and holds great potential for the development of products that facilitate medical waste classification, which could eventually become widely available throughout China. (Zhou et al. 2022).

Owing to the vast volume of data in internet networks & savings and time to analyze them, the use of feature reduction algorithms can be very useful. Kurniabudai et al. Have developed a method that uses Information Gain. Information Gain Selects the reduction of dimensions by selecting the most interrelated features among the features. Random forest algorithms NB, J48, RT, BN are used to select the feature. The results show that the random forest with a selection of 22 interrelated features has attained the highest rate of 99.86% and also J48 with selection of 52 features has reached an accuracy of 99.87% which has a longer execution time (Stiawan et al. 2020).

Abrar et al. Have introduced approach that includes 4 steps. The first step is data preprocessing. Step Two 4 basic subsets of model evaluation are identified. Step 3 Several classification ML utilized for testing and training. In the fourth step, the results are evaluated with several different parameters. The selection of features from the dataset is random. Randomly selecting features is an efficient approach to reduce model computation time and complexity. MLP, LR algorithms have been used for testing (Abrar et al. 2020).

Meyer and Labit in their research used two methods of signature-based and pattern-based forecasting. First, 4 prediction algorithms are combined and then the prediction of

network traffic behavior is matched to the pattern. 4 machine learning algorithms are used. Initially, NB was used for normal traffic due to low time and good detection. Decision tree and random forest are good for detecting Probe, R2L, U2R. Finally, Gradient Boosting is used due to the training time and good accuracy. (Meyer and Labit 2020).

Zhau et al. used a feature selection-based framework to solve intrusion detection problems. In the initial phase, a HA known as CFS-BA is introduced to reduce data dimensions, which selects a subset based on related features. Then a group approach is introduced which is a combination of C4.5, RF, ForestPA, and finally the learner-based probabilistic distribution to detect attacks. In this paper, N- KDD, A-ID, CIC-IDS2017 have been used for testing and testing. Ultimately, it has been determined that the proposed approach demonstrates superior efficiency compared to other approaches. (Zhau et al. 2020).

Chkirbene et al. Have suggested hybrid approach that combines two ML algorithms. In this system, Random F algorithm is used to select the feature and CART algorithm is employed for classifying various attacks. The suggested approach has been evaluated using the UN-NB data set. The findings demonstrate that the suggested method outperforms other approaches. (Chkirbene et al. 2020).

Elmrabit et al. In presenting their method, they have considered the problem of intrusion detection by classifying network traffic. For the experiment, simulated network traffic and experimental data containing attacks were used. In this paper, 12 ML algorithms for network AD are evaluated. Evaluation was performed in three datasets CIC-2017 and UN-NB15 and datasets obtained from experimental work. The evaluation the results demonstrate that the RF algorithm exhibits superior performance compared to other algorithms. (Elmrabit et al. 2020).

Thaseen et al In their method. Have detected intrusion through packets in the network with the aim of classifying them without decryption. This method is done using ML.

This article utilizes the Wireshark tool for analysis. ML algorithms such as: DL are used in this paper (Thaseen et al. 2020).

Enigo et al. devised a hybrids realtime IDS that can detect intrusions. This article uses the NSL-KDD dataset. The objective of this paper is to construct a three-tier IDS that can detect new attacks. Snort is an open source IDS based network that uses an incoming traffic signature and then compares it to a malicious signature. Snort IDS is used as the first layer and checks if network traffic is malicious. The second layer is network anomaly detection, which includes a machine learning model that uses supervised algorithms. The third layer categorizes the generated alerts (Enigo et al. 2020).

Conventional ML are not effective in detecting intrusion, instead a hybrid solution with several models can have good results. Suggested model, proposed by Rajagopal et al., Is a supervised learning algorithm that uses a framework with the concept of stack generalization to classify. Analysis and classification by combining SVM algorithm and other algorithms such as J48, AdaBoost, Random Forest, BayesNet have better results than SVM which performs intrusion detection without combining other algorithms. The purpose of the proposed method is to obtain a reliable prediction that uses a group technique called stacking (Rajagopal et al. 2020).

Waskle et al. Have put forward an IDS development method using Principal Component Analysis and stochastic forest algorithm. Principal Component Analysis was employed to decrease the size the input dataset and random forest was used for classification. The findings indicate that the suggested approach outperforms alternative approaches in terms of results. (Waskle et al. 2020).

Arivardhini et al. Suggested a hybrid IDS that combines Support Vector Machine, decision tree (J48) and NB algorithms. This approach aims to minimize the false (FP). The N-K data set was utilized for training and testing purposes. After balancing data set, it shows good results and improves accuracy (Arivardhini et al. 2020).

Bachar et al. Suggested methodology intrusion detection behavior that is able to detect novel attacks without utilizing endorsement data base. In this paper, the Ml utilized for classification, employing both P & G properties. Model evaluated using the UN-NB15 data set. Promising results were achieved in terms of (DR) when compared to other classification patterns, including MLP, RF, ANN, and RepTree. (Bachar et al. 2020).

Saing et al. Created an AD system. In this method, the fusion of (SVM) algorithms and DT and NB are discussed. The outcomes indicate that at the Support vector machines algorithm has a good performance in feature classification (Saing et al. 2020).

Previous work detects known attacks such as Dos, Probe, U2R, R2L. There are many attacks that are not yet known and should be identified. The proposed method tries to identify ZD and unidentified attacks. Most previous work has also used covers fewer attacks than the UNSW-NB15 dataset. Both data sets will be used in the proposed method.

Table 2.3 Comparison of previous studies

Algorithm		Data set	Type of Detection	Attacks covered	Detection accuracy (%)	Detection rate (%)	Source
Random forest-SVM	With 14 features	KDD99	-	Dos- R2L- U2R- Probing	-	93	[36]
	With 41 features					90	
Random forest, decision stamp,Stochastic Gradient Descent , ,Naïve Base		KDD-Cup99	-	Dos- R2L- U2R- Probing	99.76	-	[37]
LogitBoost, Random forest		UNSW-NB15	supervised	Browser- Brute-force- DoS- Scan- Back-door- DNS	99.99	-	[38]
Random forest		Kyoto2006+	supervised	-	99	-	[39]

Table 2.3 Comparison of previous studies continue

Algorithm	Data set	Type of Detection	Attacks covered	Detection accuracy (%)	Detection rate (%)	Source
Random forest	NSL-KDD	-	Dos- R2L- U2R- Probing	99.83	-	[11]
K-Means						
Random forest - Gaussian clustering				99.84		
Random forest	KDDCup 99	-	Dos- R2L- U2R- Probing	99.94	-	[41]
J48 ,Naïve Base,REP Tree						
NB, KNN, LR, DT	NSL-KDD	-	Dos- R2L- U2R- Probing	99.95	-	[42]
Random forest, SVM	NSL-KDD	-	-	91.7	-	[43]
	UNSW-NB15			91.8		
Stochastic Gradient, Random forest , logistic regression , SVM And a tidy model	NSL-KDD	-	Theft- Disclosure- Dos	99.99	-	[10]
Random forest	NSL-KDD	supervised	Dos ,Probe, U2R, R2L	99.12	-	[3]
Random forest	NSL-KDD	supervised	Probe, DOS, U2R, R2L	99.8	-	[46]
Decision tree						
SVM, KNN, …						
AdaBoost	CICIDS2017	supervised	DDOS	97.34	-	[47]
Decision tree,				94.68		
Random forest	CICIDS2017	supervised	DOS, Port Scan, Web Attack, Brute Force, …	99.86	-	[48]
J48				99.87		
Random forest	KDD99	supervised	Probe, Dos, U2r, R2L	-	98	[51]
	NSL-KDD	-	Probe, Dos, U2r, R2L	99.81	99.8	[52]
CFS-BA, Random forest	AWID	-	flooding, impersonation, injection	99.52	99.5	
	CIC-IDS2017 (Wed.)	-	DDoS, BS, XSS, SQL Injection, Infiltration, PS, and BN	99.89	99.9	

Table 2.3 Comparison of previous studies continue

Algorithm		Data set	Type of Detection	Attacks covered	Detection accuracy (%)	Detection rate (%)	Source
Random forest ، CART		UNSW-NB15	supervised	Analysis, Backdoor, DOS, Shellcode, and Worms attack	95.73	-	[53]
		KDD99		classes	97.3		
Random forest		CICIDS-2017	supervised	DoS, PS, SQL injection, BF	99.9	-	[54]
		UNSW-NB15			87.7	-	
SVM		UNSW-NB15	supervised	DOS, Fuzzers, Backdoors, Analysis, Generic, Reconnaissance, Shellcode, Worms	94	-	[55]
SVM	Dos	NSL-KDD	supervised	Probe, Dos, U2r, R2L	98	-	[1]
	Probe				98		
	U2R				99		
	R2L				99		

2.10 Chapter Summary

In this chapter, the literature and background of the subject were examined. Internet security is very important. There exist a multitude of methods for detecting intrusions on networks, one of which is machine learning and classification algorithms. Therefore, in this chapter, these cases were thoroughly examined.

3. METHODOLOGY

3.1 Introduction

Today, the network plays a crucial role in modern human life. Internet and Internet networks have a direct impact on much of human life. Extensive technologies have emerged in connection with the Internet. Along with the increasing advances in human life made using the Internet, there are also threats. One of the major threats to Internet networks is penetration.

The basics of the research were stated in the first chapter; in the second chapter, the literature were reviewed and in this chapter, the specifics of the proposed approach will be elucidated

In this chapter, the following steps will be tried:

- Provide an overview of the work

- Explain how to use feature selection tools

- Explain meta-heuristic for feature selection and nonlinear optimization of pattern parameters

- Explain how to use DT, RF and DNN algorithms to discover intrusion

Each of these cases will be described below.

3.2 General Procedures of the Proposed Method

In today's digital landscape, the detection of attacks and anomalies has emerged as a significant concern and a challenging task on the Internet. As the utilization of Internet infrastructure continues to expand across various domains, the frequency and severity of threats targeting this infrastructure have also escalated. Various kind attacks, containing DOS assaults, data manipulation, unauthorized access, malicious activities, scanning activities, espionage, and misconfigurations, pose substantial risks to Internet-based systems, potentially leading to system failures and disruptions. DoS attacks aim to conquer a target submerge a system or network by inundating it with surplus volume of flood a system or network by overwhelming it with an abundance of traffic or resource requests, causing it to be incapable of addressing valid user demands. Data manipulation attacks involve unauthorized modification, deletion, or extraction of sensitive information, compromising the integrity and confidentiality of data. Unauthorized access attempts and malicious activities can result in unauthorized control or exploitation of systems, potentially causing extensive damage. Additionally, scanning activities conducted by attackers involve probing a network or system to identify vulnerabilities or weaknesses that can be exploited. Espionage attempts involve unauthorized surveillance or data gathering for illicit purposes, while misconfigurations can leave systems vulnerable to attacks or because unintended consequences. Given the critical role of Internet-based systems in various sectors, including finance, healthcare, and communication, the impact of these attacks and anomalies can be significant. Therefore, the development of robust and efficient detection mechanisms, including IDS and algorithms designed to detect anomalies, is essential to safeguard these systems from potential threats and mitigate the risks associated with attacks and anomalies on the Internet (Hasan et al. 2019). Therefore, these attacks and abnormalities must be identified with the help of methods. One of the most effective ways to diagnose seizures and abnormalities is to use machine learning tools and classifiers. In the following, while explaining ML and enumerating a few examples of their applications in this domain will be provide .

"Optimizing Performance Based on Data Samples and Past Experiences" provides a description of machine learning in the Alp Aydin language. This expression provides a simple definition of machine learning. In ML, data role an undeniable play. ML algorithms are employed to unveil the underlying pattern, knowledge and properties of data. The performance of machine learning algorithms can be influenced by the quality and content of the data they are provided with. (sah et al. 2020). Machine learning has different branches and fields. One of the important areas in machine learning is classification or categorization. Knowledge classification structures provide us with many important sources of data structure. Classification is referred to as the procedure of assigning incoming data to predetermined classes. In this scenario, a classification model, established on the gathered data, is allocated to one of the classes.(Alpaydin 2010).

Based on the cases mentioned in the previous paragraphs and considering the high variety of attacks on the Internet and also the various algorithms that exist within the realm of ML, in the following, we try to express a more accurate process of this research. In this research, we intend to examine various attacks in the field of Internet and Internet in general, such as DoS and DDOS, Probe, U2R, etc (Sai Kirana et al. 2020).

To do this, traditional classification tools and feature selection based on meta-heuristic optimization algorithms will be used. Also, considering that deep learning has provided acceptable results in many areas in recent years, deep learning will be used to classify and diagnose attacks. Also, due to the fact that ML for instance DNN and SVM have parameters to be adjusted, in this study we try to optimize the parameters of these algorithms while selecting the feature. Accordingly, The overall structure of the proposed method will be as outlined below, encompassing two fundamental steps:

1- Simultaneously optimizing the parameters of the machine learning algorithm while selecting effective features.

2- Making the classification model using the output of the first step

Figure 3.1 highlights the key characteristics of the suggested method

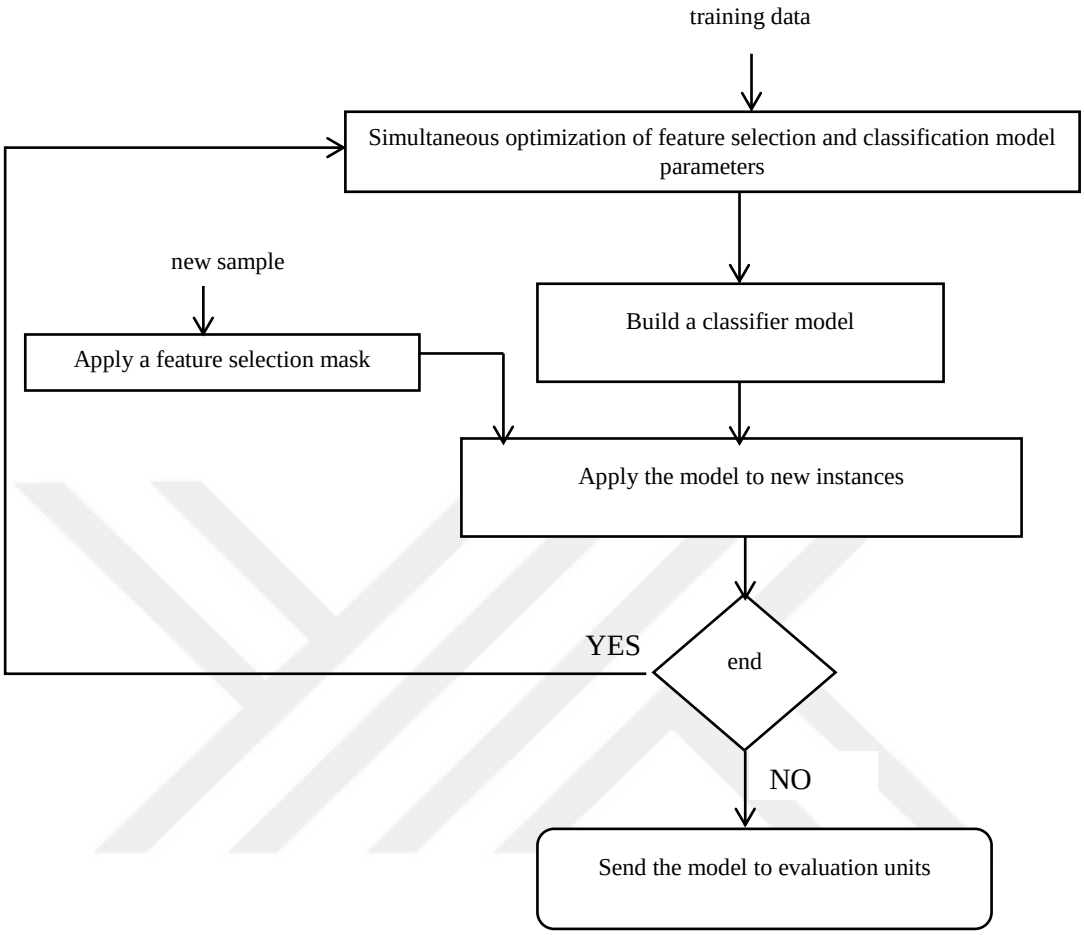


Figure 3.1 Framework of the suggested approach

According to Figure 3.1, the proposed method has 4 steps, the main steps of which are located in boxes 1 and 2. Feature selection in this study is done by wrapper method. The wrapper method in this study evaluates the selected features using a model constructed by a machine learning model (traditional classifiers).

As mentioned, to select the feature and optimize classification model simultaneously; the metaheuristic algorithm is employed. In this research, the multi-objective optimization algorithm will be used. Figure 3.2 illustrates the overall pattern of the suggested approach.

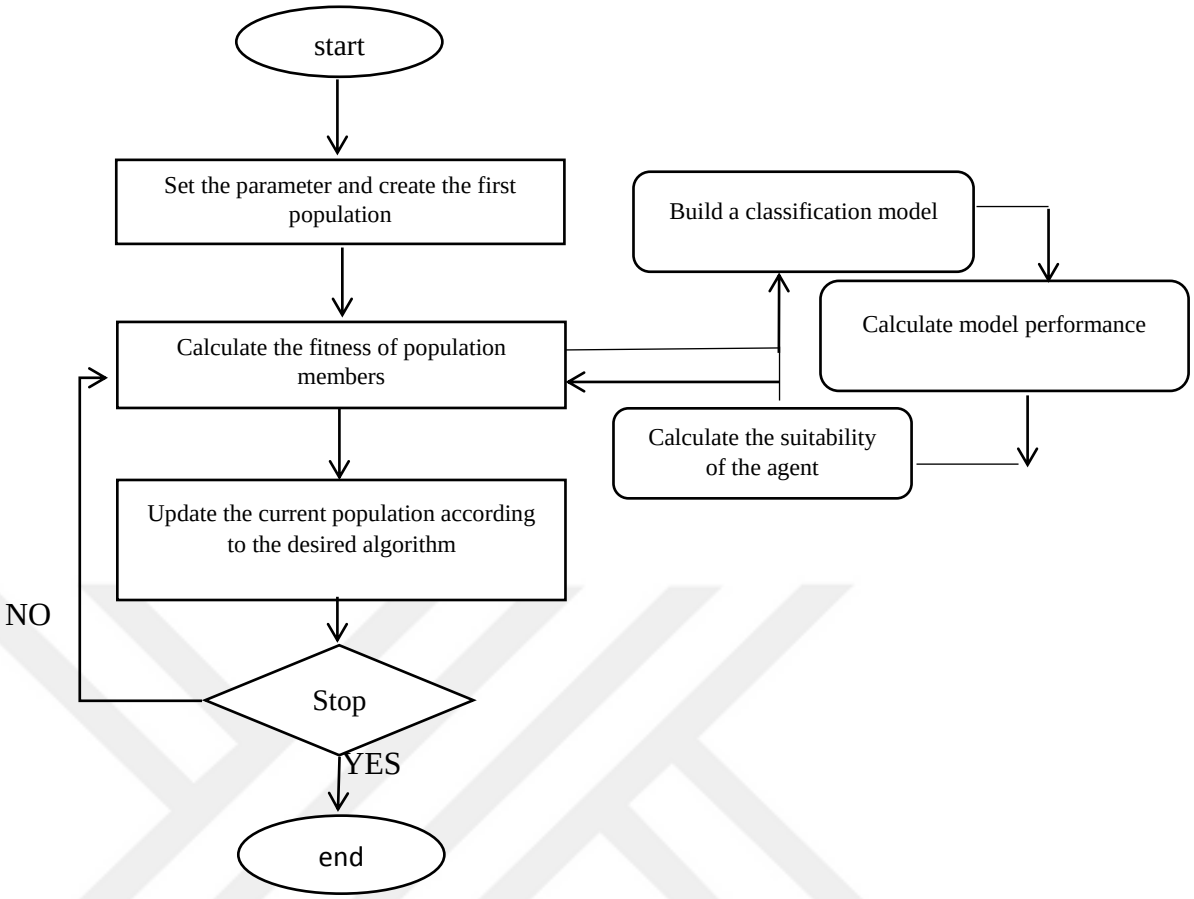


Figure 3.2 Diagram depicting the flow of the suggested approach

To calculate the suitability of each method, a classification model tailored to each factor is used. Accordingly, the following traditional classifiers are used in this study:

- ANN
- SVM
- RF

The following representations will also be used to classify in depth learning:

- CNN
- RNN
- LSM
- GRU

Combining meta-heuristic algorithms of GW[23] optimization algorithm and NSGAII[24], MOPSO[25] to optimize learning model parameters and feature selection

-Recommended Method: Within this particular section, the suggested approach will be examined. The suggested approach consists of four steps, which are illustrated in Figure 1. The core concept behind this approach is to identify the most impactful features within a given database and incorporate them into the final layers of CNN architectures. By doing so, the objective is to enhance the accuracy of the model.

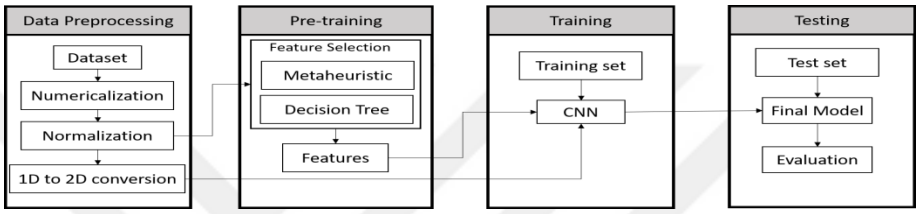


Figure 3.3 Proposed method to combine metaheuristic algorithms and CNN for intrusion detection

The four stages are data preprocessing, pretraining, training, and testing. The ultimate object is to train a CNN model that will be used to detect the attack in real time.

3.3 Objective Optimization Algorithms

Section delves into an exploration of the concept the concepts of optimization and multi-objective optimization, as well as the process of solving these types of problems. Optimization refers to the task of finding the optimal a resolution to a problem by considering a collection of conditions and limitations. In such cases, it is necessary to either maximize or minimize the outputs of the goal functions. In multi-objective problems, the scenario differs from single-objective problems. Instead of having a scalar objective function, we deal with a vector of objective functions. The solution space for multi-objective problems is not arranged in a single objective state. For instance, if Equation 3.1 is defined for a multi-objective problem, we would have:

[23] Grey Wolf
[24] Non-dominated Sorting Genetic Algorithm II
[25] Multi-Objective Particle Swarm Optimization

$\min_x f(x)\ \ f: X \to R^m\ , f = (f_1, f_2, \dots, f_m)$ (3.1)

That is, the function f itself has m components and is a dimension m vector in where m represents the count of goal functions andR^m is the Cartesian product of R times m times. So space is no longer an orderly space. Because comparative operators such as smaller (<) or larger (>) in this space cannot be fully defined .The problem with the answer space in multi-objective problems is that it is not sequential, which can be solved in two ways (Agrawal et al. 2021).

- Converting a goal problem into a single multi-objective problem (analysis) or weighting before solving

In these methods, the problem space becomes a scalar space. These methods are also called pre-dissolving weights. In these methods, decisions about the preferences of objective functions are presented before solving (before searching for the answer space) and weights are usually defined by experts in the field. The answers are completely influenced by the experts. Object functions are transformed into a single-valued goal function by different methods and the problem is solved as a single objective. These methods include weighted sum, metric weight methods, ideal planning, etc. In these methods to achieve bringing a set of effective answers (answers) .It is essential to iterate the algorithm multiple times, aiming to obtain a fresh solution with each iteration (Qin et al. 2008).

- Direct solution: In these methods, a criterion for the superiority of the answers in the multidimensional space of the objective functions is defined and the problem is solved based on that criterion in a multi-objective manner. These methods are also called post-solution weighting. In fact, the problem is completely solved in several ways, the answers are obtained and then the final answer is obtained according to the preferences announced by the decision maker (Mirjalili and Lewis 2016). In other words, there is not only one optimal answer, but the answer is a set of points that can be a level, a curve and 2, which is called the trade-off answer curve or effective answers. In this method, no initial weight is entered into the calculations. In MOO the objective is to discover a

set of optimal solution, taking into account all the objectives. Therefore, the user can make a choice using higher level qualitative considerations.

For an ideal MOO process, the following steps required performed:

1. Find several optimal give-and-take answers, with a diverse range of values for various purposes.

2- Choosing one of the optimal solutions by utilizing higher-level notice.

In Figure (3.3), multi-objective optimization process is shown as weighting after solution (Haykin 1994).

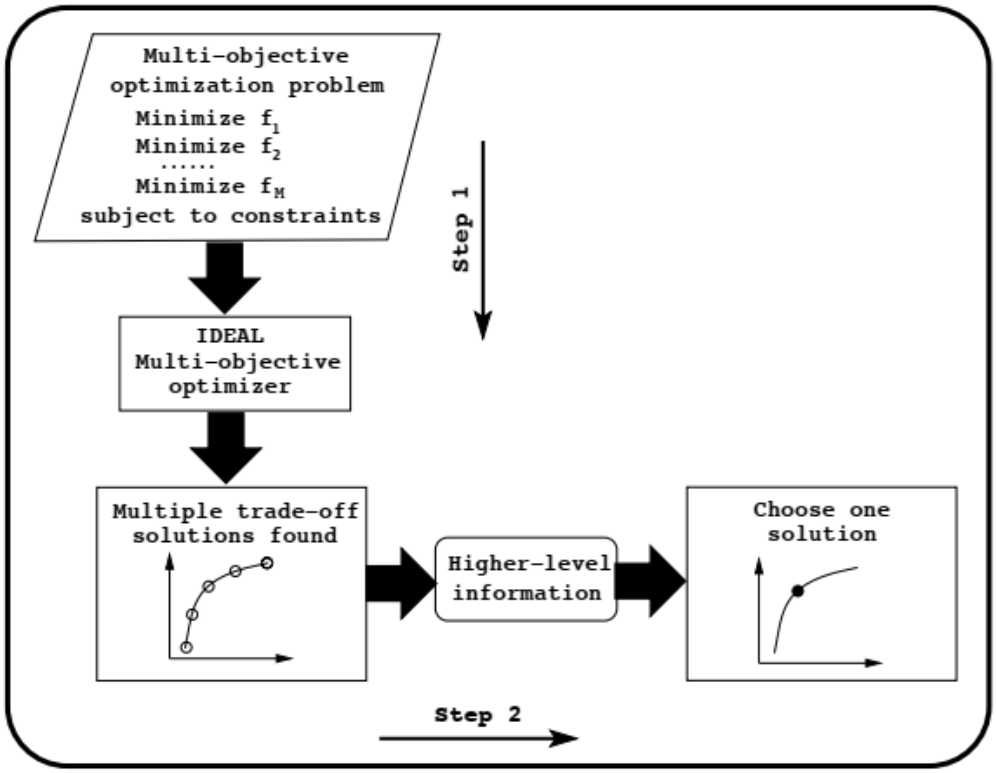


Figure 3.4 Multi-objective optimization procedure for weighting after solution

-The concept of dominance

The primary challenge in MOO is that it is rare to discover a single response optimizes all goal simultaneously. Instead, we aim to identify a set of response points known as

the effective response front, where none of these points dominate the others. These points on the front represent trade-off solutions.

To determine the effectiveness of a solution set, we establish a superiority criterion. Answer x_1 is considered superior to answer x_2 if two conditions are met:

Answer: x_1 does not exhibit inferior performance compared to x_2 in any of the goal.

Answer: Answer x_1 surpasses x_2 in at least one goal.

In other words, x_1 is at least equivalent to x_2 in all goal and outperforms x_2 in at least one goal. By establishing and comparing these criteria across various resolutions, we can identify the effective answer set, which comprises ND or PO candidates.

The aim of solving a MO issue is to provide decision-makers with a comprehensive set of effective potential represent varying trade-offs among the aim. This allows decision-makers to investigate the solution domain and pick the most suitable solution based on their preferences and requirements, considering the trade-offs involved.

$$f_i(x_1) \geq f_i(x_2) \; for \; all \; i = 1,2, \dots, M \tag{3.2}$$

Answer x_1 is definitely superior to answer x_2 in at least one goal, or:

$$f_j(x_1) > f_j(x_2) \;\; For \; at \; least \; one \; j \in \{1,2, \dots, M\} \tag{3.3}$$

Figure 3.4 shows the case in which the optimization problem consists of two objectives, one for minimization and the other for maximization. In this form, answer A prevails over answer C; Because it has obtained a smaller value for both objective functions. But in the case of answers A and B, neither can prevail over the other. Answers B and C can also be defeated without the presence of answer A (Alzubi et al. 2019).

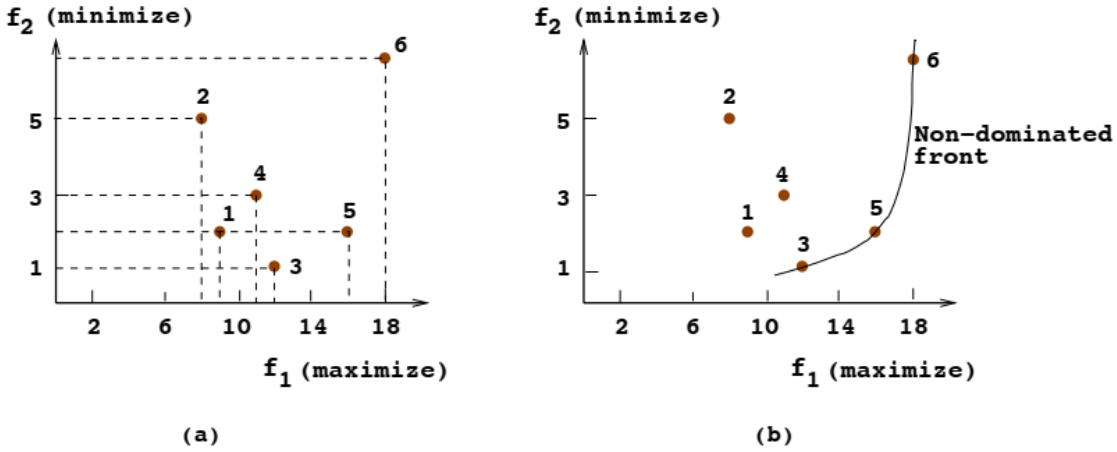


Figure 3.5 An example of an answer defeated and undefeated

Among the sets of answers *P*, the set of invalid answers P ^ 'are those that aren't defeated by any means of the elements of the set referred to as invalid.

3.3.1 NSGAII[26]

This algorithm also uses congestion distance to obtain a more uniform response front than other algorithms and to estimate the density of points around the answers. It should be noted that the swarm distance factor is used to better select the answers in terms of dispersion on a front (Verma et al. 2021). According to Figure 3.6, in the case of two targets, the congestion distance for the ith response is calculated by the following equation:

$$d_i^1 = \frac{|f_1^{i+1} - f_1^{i-1}|}{f_1^{max} - f_1^{min}} \quad (3.4)$$

$$d_i^2 = \frac{|f_2^{i+1} - f_2^{i-1}|}{f_2^{max} - f_2^{min}} \quad (3.5)$$

$$d_i = d_i^1 + d_i^2 \quad (3.6)$$

[26] Non-dominated Sorting Genetic Algorithm II

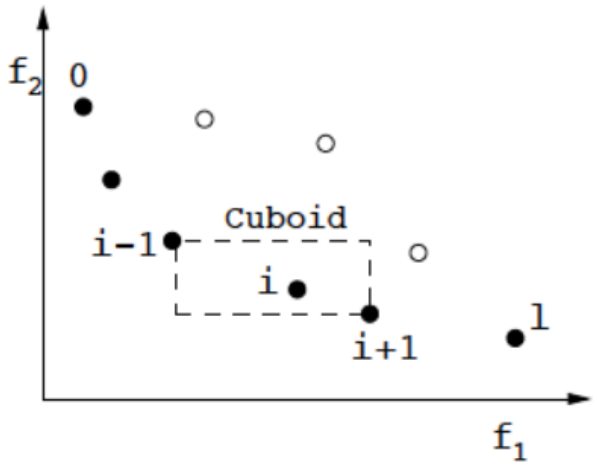


Figure 3.6 The congestion distance of the response in NSGA-II

To calculate the congestion distance in general, the following equation is used:

$$d_i^j = \frac{\left|f_j^{i+1} - f_j^{i-1}\right|}{f_j^{max} - f_j^{min}} \tag{3.7}$$

$$d_i = \sum_{j=1}^{m} d_i^j \tag{3.8}$$

- NSGA-II algorithm steps

Once the initial population is created, the fitness criterion calculation begins, where the objective function values are evaluated for each individual. The population is then sorted based on the concept of domination.After applying dominance-based sorting to the answers, the next step is to sort them based on their swarm distance. This additional sorting step helps create further order among the optimal answers, providing a clearer representation of the trade-offs. Following the sorting process, parent selection is carried out using the multi-objective tournament selection method. This method involves selecting individuals from the population based on their ranking in lower ranks (with a better answer being assigned a lower numerical rank). The guarantees that individuals with superior physical condition have an increased probability of being chosen as guardians. Within the multi-objective tournament selection, a member is

selected who possesses a greater crowding distance. The crowding proximity is a metric that assesses the extent to which an individual is dispersed within its immediate vicinity. By favoring persons having a greater spacing disparity, diversity is maintained within the selected parents, promoting exploration of the solution space and preventing premature convergence to a suboptimal solution.

In summary, the process involves fitness evaluation, dominance-based sorting, swarm distance sorting, and finally, parent selection using the multi-objective tournament selection method with consideration for crowding distance. This iterative process helps drive the evolutionary algorithm towards finding a diverse and effective collection of resolutions regarding the multi-objective optimization challenge in question. (Mirjalili et al. 2016).

Intersection and alterations are subsequently executed to produce new offspring. In the subsequent phase, the original community and the community derived from the overlap & mutation are first classified based on hierarchy and a few of those that possess a superior-ranking ones are eliminated. The entire process is reiterated until the intended generation. Figure 3.7 illustrates the progression of the NSGA-II algorithm.

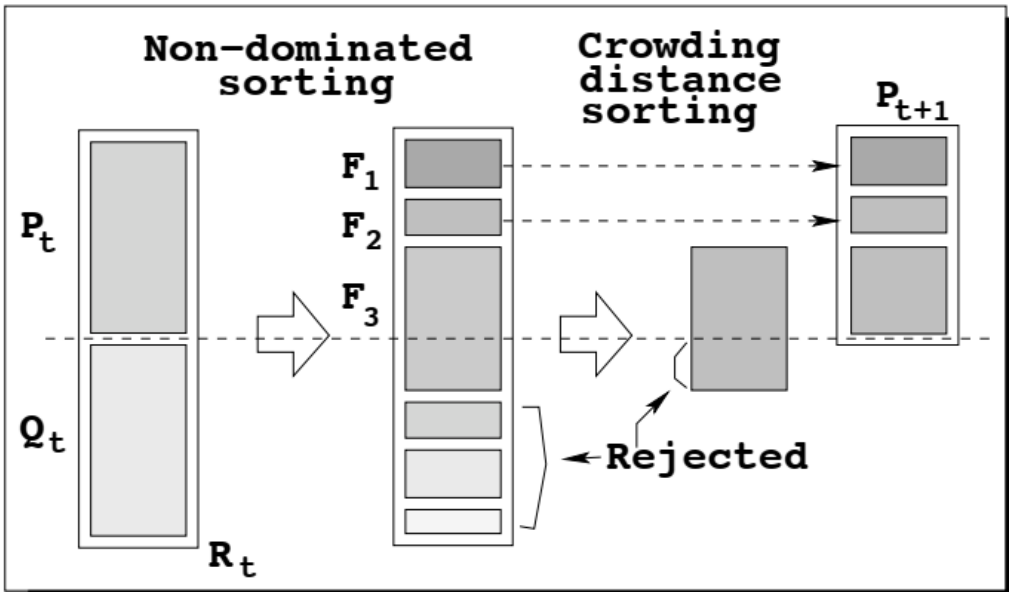


Figure 3.7 NSGA-II process

Appropriate data sets will be used to better measure these methods. Which will be examined in the fourth chapter. The model will also be evaluated for both class and multi-class classification modes.

3.3.2 MOPSO[27] algorithm

One of the meta-heuristic optimization techniques is the particle swarm optimization algorithm. It is first used an optimization method in 1995 (Yang et al. 2009). The algorithm is based on birds' behavior. Requiring only a few computational operators, the algorithm is easy to the implement and really cost-effective. In addition, some cases can overcome the problems that we may encounter when using the genetic algorithm.

PSO is a multi-factor search method whose agents operate in parallel. Particles in PSO are conceptually entities that fly in multidimensional search space. In PSO, each particle has its own displacement speed and position. The parameters of this algorithm are:

V_max: The maximum speed that each particle within the exploration area can have. The permissible speed of every individual particle is in the range $(-V_{max}, V_{max})$.

repose weight coefficient (ω): indicates the tendency of the particle to maintain its current position.

Parametersc_1 and c_2, respectively, indicate the effect of the best state of each individual particle and the optimal state of the group of individuals (based on what has been observed so far) on determining the new position of each particle. $c_1$is called self-confidence and c_2 is called community confidence.

• MOPSO

The multi-objective particle cluster optimization algorithm, introduced by Coelho in 2004 (Yang et al., 2009), is an extension of the (PSO) algorithm designed specifically

[27] multi-objective particle swarm optimization

for solving multi-objective problems. One key addition in the MOPSO algorithm is the concept of an archive or repository, which enhances the capabilities of the PSO algorithm. In the MOPSO algorithm, selecting the best overall answer (G-best) the individual's best personal memory are crucial steps in achieving effective multi-objective optimization. When a particle initiates a move, it chooses a leader from the repository. Rector should be a part of the collection while also being superior to other members in the repository. Symbolizes or embodie the Pareto front and contains non-dominated particles. Unlike in the PSO algorithm, where there is only one global best solution for a single objective, the MOPSO algorithm incorporates a repository that holds several invincible and well-fitting particles that make up the set of effective solutions. By utilizing the repository in the MOPSO algorithm, multiple trade-off solutions can be maintained and considered for the optimization process. This enables the algorithm to explore the solution space more comprehensively, capturing a diverse set of solutions that balance different objectives. The concept of the repository enhances the performance of the PSO algorithm in tackling multi-objective optimization problems, allowing for a richer and more informed decision-making process. (Alzaqebah et al. 2021)

To contrast the finest array of individual recollection, we undertake the subsequent actions:

- In the event of the novel placement overcomes the utmost remembrance, then the fresh location supplants the superior recollection. In algebraic expressions:

$$Pbest_i^{n+1} = X_i^{n+1} \tag{3.10}$$

- In the event of the new circumstance being vanquished by the superior recollection, nothing will be done, in mathematical terms:

$$Pbest_i^{n+1} = Pbest_i^n \tag{3.11}$$

If neither of them defeats each other, we consider one as the best position vector. This selection is random (Elmasry et al. 2020).

The order of execution of this the algorithm is outlined as follows:

1) Determine the parameters required to run the MOPSO algorithm: such as maximum iteration to run the algorithm, population size, values C1, C2 and the amount of tank members
2) Create an initial population
3) Initialize the individual finest solution for every particle-centric on its initial position and evaluate its fitness
4) Separation of defective members of the population and storage in the reservoir
5) Each particle selects a leader from among the members of the tank and performs its movement, updating its speed and position.
6) Update the best personal memory of each particle
7) Add new defective members to the repository
8) Remove the defeated members of the tank
9) Upon reaching the termination condition is not satisfied, the algorithm is repeated from step 5.

Another optimization algorithm will be discussed below.

3.3.3 Gray wolf multi-objective optimization algorithm

GW is top-level carnivores that fulfill a crucial play in ecosystems as top predators. They are known for their social nature and often form packs or groups for various advantages, including hunting, defense, and social interaction. The size of gray wolf packs can vary, but on average, they consist of five to twelve wolves. However, pack sizes can range from as few as two individuals to larger groups of up to 30 or more, depending on factors such as prey availability, habitat conditions, and pack dynamics. The pack structure typically includes an alpha male and an alpha female, which are the dominant breeding pair, along with their offspring and sometimes other subordinate

individuals. (Devi et al. 2017). An interesting point about the lives of these predators is the existence of a strict hierarchical social superiority shown in the figure below.

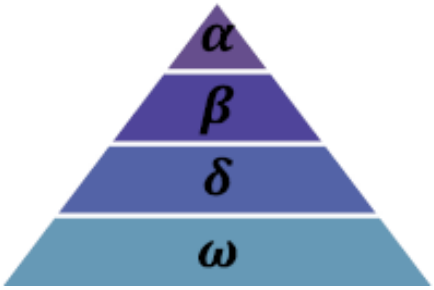


Figure 3.8 Hierarchy in gray wolves

The alpha pair determines the location of hunting, sleeping and waking. Beta wolves form the second category of this pyramid and help alpha in decision making. Beta wolves are the best option at the time of alpha death or when alpha is very old. Omega wolves are at the bottom of the hierarchy and play a pre-death role. Omega wolves are subordinate to other wolves and eat the last of all. Any wolf that does not exist in this structure is called a DW. DW is lower than α & β wolves and higher than gamma canines.

In addition to the social hierarchy within the flock, mass hunting is also an interesting feature of gray wolves. The main steps of hunting these wolves are as follows:

Search, chase and approach the hunt

Siege and exhaust the prey with successive attacks until the quarry ceases its motion

The ultimate assault during the pursuit. As mentioned before, during hunting, gray wolves surround their prey. Hunting mechanism is modelled as follows:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{3.12}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \tag{3.13}$$

t represents the repetition number, vectors A and C, coefficient magnitudes, X_p the prey location vector, and X the gray wolf location vector. Vectors A and C are calculated as ensues:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \tag{3.14}$$

$$\vec{C} = 2.\overrightarrow{r_2} \tag{3.15}$$

A decreases progresses uniformly from 2 to 0 throughout the successive cycles, and r_1 and r_2 are random vectors in the range (1 and 0).

To simulate how gray wolves hunt, it is postulated that α, β and delta exhibit superior information about the hunting position. As a result, the top three answers are obtained so that this part of the work is saved and the other wolves are forced to update their position according to these top three answers. The following relationships are stated for the task:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right|, \overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}\right|, \overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right| \tag{3.16}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.\left(\overrightarrow{D_\alpha}\right), \overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.\left(\overrightarrow{D_\beta}\right), \overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.\left(\overrightarrow{D_\delta}\right) \tag{3.17}$$

$$\overrightarrow{X}\ (t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \tag{3.18}$$

Gray wolves pounce on their prey upon its cessation of movement. To model this behavior and approach the targeted, the magnitude of aundergoes a decrease and by decreasing the value. The range of variations in A is likewise diminished. In other

terms, A is an arbitrary value within the interval [-2a, 2a] that changes from two to zero in repetitions. When the values of A are in the range [-1, 1], the wolf's next location can be in any situation amidst the present whereabouts and the prey location. While searching for prey, the wolves move away and gather again when attacked. In mathematical modeling, a search operation is performed for random values greater than one and less than negative one for A.

Another factor of this algorithm that has an effect on the search section is the C vector. This vector, according to the definition given, contains random values in the range [0, 2]. This array supplies arbitrary data for the purpose of make significant (C> 1) or minor (C <1) the impact the prey position in determining the separation (D). This allows the algorithm to behave more randomly and avoid local optimization. The vector C, unlike A, has a random value during all iterations and its value is not decreasing.

3.4 Feature Selection and Optimization of Classification Model Parameters

Based on the above, this section will show how to choose the characteristic and optimize the parameters of the categorization algorithms such as SVM, NN and RFs. (Hasan et al. 2010). Feature selection and optimization are done using the algorithms mentioned in Section 3.3.

3.4.1 Representation of factors for feature selection and parameter optimization

Problem solving agents in the meta-heuristic algorithms used are shown in Figure 3.9.

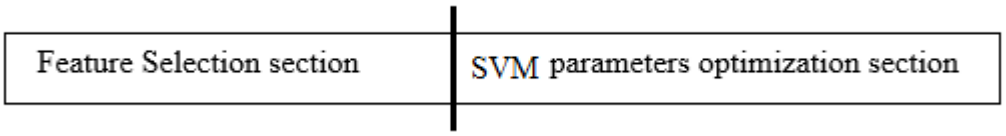


Figure 3.9 General form of problem solving agents in the proposed method

Given that in this study the purpose is for picking the characteristic and fine-tuning the parameters of the categorization methods simultaneously, so the problem-solving factor in this study, as shown in Figure 3.9, has two parts, the first part (left) is related Select the feature and the second part (right) correlates with the optimization of classification parameters such as SVM. For example, for SVM, in this study, three parameters, C (penalty coefficient), γ (RBF kernel parameter) and K (kernel function) are optimized. For other classifiers, these parameters will be described in Chapter 4. Diagram 3.10 illustrates a sample of the factor representation used in these algorithms:

f_1	f_2	f_3	...	f_i		f_N	*SVR Parameters*		
0.5	0.2	0.3	...	0.75	...	0.6	28.5	0.05	1.6
w_1	w_2	w_3	...	w_i		w_N	C	γ	ε

Figure 3.10 An example of a problem-solving agent in the proposed method

According to Figure 3.10, the magnitude of each factor corresponds to N + m, where N signifies the count of attributes per instance in the dataset, and m denotes the number of parameters of the classification algorithm necessitating optimization. The point to be noted is that in choosing the attribute, the objective is the choice (existence -1) and the non-choice (non-existence -0). But in this play, to handle with a stronger structure in in terms of conveying the significance of features and to express the significance of features only in spite of having or not having them, the method of weighting the features has been used, ie each part has a factor in selecting the feature. The degree of importance of that attribute is, and the higher this weight (w_i ; $1 \leq i \leq N$), the greater the chance of selecting that attribute. Now, in order to extract useful properties and remove non-useful properties, the display of Figure 3.9 in the feature selection section should be converted to a binary vector (one = feature selection, zero = no feature selection). The following conversion method is utilized to transform the feature selection section in Figure 3.9 to a binary vector:

$$f(w_i) = \begin{cases} 1 & \text{if } w_i \geq th \\ 0 & 0.w \end{cases} \tag{3.19}$$

In the relation 3.19 w_i is equal to the weight of the ith property in the factor and th is the threshold value, if the weight of the property in the factor exceeds the threshold magnitude, that property is selected, otherwise the desired property is not selected. In this study, the threshold value was considered equal to 0.5 (th = 0.5). Therefore, using Equation 3.19 and the value of th = 0.5, Figure 3.9 is shown as Figure 3.11:

f_1	f_2	f_3	...	f_i		f_N	*SVR Parameters*		
1	0	0	...	1	...	1	28.5	0.05	1.6
w_1	w_2	w_3	...	w_i		w_N	C	γ	ε

Figure 3.11 Displays the agent in binary to specify the selected properties

The representation of agents in Figure 3.11 is used in the fit function.

3.4.2 Calculating the suitability of agents

In feature selection, we seek to remove and reduce irrelevant and duplicate features along with building an accurate model according to the chosen set of features. Given that the model presented in this study uses the wrapper method to select the feature, both concepts should be removed. Irrelevant and repetitive features as well as model accuracy should be used in evaluating each factor. Therefore, this study can be regarded as a dual-objective optimization problem. Accordingly, the definition of fitness will be defined for both one-goal mode and two-goal mode (Zhou et al. 2020)

- To calculate the suitability of the agents in the proposed method in the single-objective mode, the equation 3.10 was used.

$$\mathrm{Fitness}(\mathrm{Agent_i}) = \mathrm{W_1} * \mathrm{MP}(\mathrm{Agent_i}) + \mathrm{W_2} * \frac{\mathrm{SF_i}}{\mathrm{TF}} \qquad (3.20)$$

In the case of 3.20 MP (Agent_i), it indicates the accuracy of the model can be calculated using Equation 3.21, which is based on the performance of the model on the ith agent. In the case of 3.20 MP (Agent_i), it signifies the effectiveness of the model on

factor i. The efficiency of the model is determined by evaluating accuracy of the diagnosis:

$$Accuracy = \frac{Correct\ Detect\ Sample}{Total\ Sample} \tag{3.21}$$

In relation 3.21 Correct Detect Sample, indicates the sample count that were accurately identified and the total count sample is the total number of samples.

In relation 3.20 SF_i, the number of Selected Feature in the ith factor and TF are the total feature of each instance in the data set. The fewer features selected in a factor, the more appropriate that factor is. So the lower the TF / (SF_i) fraction, the better the factor. In In the proposed approach, the objective is to maximize the ratio of 3.20. The values $W_1$and W_2(W_1+ W_2= 1) in relation 3.20 indicate the weight of each criterion. In this relation W_1= 0.8 and W_2= 0.2 are placed.

To calculate the suitability of the factors in the proposed method in the case of two objectives, two parts of the single-objective objective function (Equation 3.20) were used. in this case:

- The main goal is to improve the efficiency of classification model, aiming for improved accuracy, precision, and overall performance in predicting class labels or outcomes. This entails creating and executing methods to enhance the model's efficiency measures, such as elevating the accuracy of correctly identified instances and diminishing instances of incorrect positive and negative classifications.

- The secondary aim is to decrease the quantity of features considered in the classification process. Feature selection or dimensionality reduction techniques are employed to identify and retain the most relevant and informative features, eliminating redundant or less significant ones. By reducing the feature space, the model becomes more interpretable, computationally efficient, and less prone to

overfitting. Furthermore, diminishing the quantity of characteristics can expedite the process of training and making inferences, particularly when confronted with extensive datasets.

- Finding a middle ground between the primary and secondary goals, as increasing efficiency should not come at the cost of sacrificing essential features or overlooking important information. Therefore, careful consideration and analysis are required to pinpoint the subset of characteristics that have the greatest impact on the classification objective while upholding exceptional performance.

3.5 Classification Algorithms

In this section, the classification algorithms used in this research are presented.

3.5.1 Artificial neural networks

ANN follows the process used by human brain to solve complicated problems. Conventional methods rely on a limited number of commands for problem solving, which lower their computational capacity; however, neural networks possess the capability to uncover patterns within data that remain elusive to human perception.

Neural networks and traditional computational approaches do not engage in direct competition; instead, they complement each other. Certain tasks are better suited for algorithmic methods, while others are better suited for neural networks. Moreover, there are challenges that demand a hybrid system combining both approaches to attain higher levels of accuracy (Hu et al. 2021).

To implement artificial neural networks using the function of the human brain, which is a biological network:

Neuron: Neural networks and traditional computational approaches do not engage in direct competition; instead, they complement each other. Certain tasks are better suited for algorithmic methods, while others are better suited for neural networks. Moreover, there are challenges that demand a hybrid system combining both approaches to attain higher levels of accuracy.

A Neuron is consisting of four primary components namely, Neurites, Cell Body, Axonal Fibers, and Junctions. Neurites are extensions that are specialized to acquire inputs from adjacent cells. They resemble a branching pattern akin to a plant and create projections that are activated by neighboring neurons and transmit an electrochemical signal to the neuronal soma. At the termination of the neurites lies a distinct biological formation known as the junction, fulfilling the function of a portal for communication. Diverse signals are conveyed to the core of the cell via junctions and neurites, where they amalgamate. The amalgamation process described earlier can be represented by a basic arithmetic addition operation. While it may not always correspond directly to addition in practice, it can be approximated as a simple summation operation within mathematical modeling. Following this operation, a specialized function is applied to the signal, and the resulting output is transmitted to neighboring cells as distinct electrical signals.

Intelligence is not the only agent in the size of the brain, otherwise elephants or whales would have to be smarter than humans. The intelligence agent of the the complexity of the human brain lies in the multitude of connections established between individual neurons throughout the organ.

The process of learning in an artificial neural network is akin to how the human brain learns. From birth, humans gradually learn by observing and experiencing various examples of problems that need to be solved, such as moving, eating, communicating, etc. Over time, we develop the ability to respond appropriately in different situations. Similarly, artificial neural networks start learning from the beginning of their formation in order to eventually provide appropriate responses to specific situations. In the domain of artificial neural networks, numerous mathematical and software models have been

suggested, drawing inspiration from the functioning of the human brain.These models are utilized to address a wide range of scientific, engineering, and other problems in various domains.

The configuration of cells in a neural network is referred to as its network architecture. The architecture involves determining the number of layers and their connections. The initial layer acquires the inputs for the network, while the final layer generates the desired outputs. In between, if necessary, there are hidden layers. The input layer receives the input signals and transmits them to the next layer based on the communication strength with that layer. The strength of communication between neurons is referred to as the weight of each neuron. Those measurements ascertain the impact of every neuron's result on the following strata of the framework. During the learning process, the network adjusts these weights through training algorithms, such as back propagation, to minimize the error between the predicted outputs and the desired outputs. By iteratively updating the weights, the network learns to make more accurate predictions or classifications based on the given inputs. Overall, the network architecture, including the number of layers and the connections between them, plays a crucial role in the education and proficiency of simulated neural networks. The measures allocated to the neurons inside the framework establish the intensity of their impact and their contribution to the network's comprehensive outcome. (Hu et al. 2021).

Middle layer: The quantity of intermediary strata and the quantity of neurons is random. The intermediary strata must be cautiously chosen to generate the appropriate outcome.

Output layer: An additional cluster of neurons also shapes the external environment through their generated results.

The neural network operates as a mathematical function that takes input from a specific quantity of input neurons and produces output based on the number of output neurons. Various types of neural networks can be identified, including:

- Multi-layer perceptron
- Hopfeild net
- Kohonen feature Map

Adaptive Resonance Theory

The subsequent chart displays the correlation between the simulated neural network and the biological neural network:

Table 3.1 Correlation between artificial and natural neural networks

Artificial neural network	Bio neural network
Neuron	Predication
Input	Dendrite
Output	Axon
Weight	Synapse

In this study, neural network parameters are optimized as follows:

- Quantity of neurons in the input stratum (feature selection)
- Quantity of intermediary strata
- Quantity of neurons in each stratum
- Rate of learning
- Training algorithm
- Stratum transition function

Furthermore, deep learning relies on the neural network, and its parameters will undergo optimization.

3.5.2 Random forest

RF is type ensemble algorithm comprising MDT. The stochastic forest classification method has significantly advanced by constructing a group of trees and performing a voting process to determine the majority vote. In the construction of random forests, two crucial elements are the bagging technique and random feature selection at each node. Here are the defining attributes of a basic random forest. (Devi et al. 2017).

- Enveloping technique: Enveloping technique by L. Briman 1996. This method is a metaheuristic rooted in self-starter and combination concepts to improve ML. Group ML algorithms combine numerous feeble apprentices to accomplish a potent apprentice. This method prevents over-fitting of data. In enveloping, favorable outcomes are generated when the fundamental categorizations of educational algorithms are unsteady (such as choice shrubs or neural meshes), thereby causing minor modifications in instructional data to result in substantial alterations in the pattern constructed by said algorithm.

The process of the begging algorithm s as subsequent; contemplate a collection of instructional D of size m. By uniformly sampling and replacing samples from D, Bagging produces n new training sets Di with initial size m. Substitution sampling permits certain specimens to be duplicated in each Di. This form of sampling is recognized as self-replicating sampling. The outcome of n model amalgamations is acquired through mean averaging for regression tasks and majority voting for classification tasks. Consequently, by re-sampling and replicating diverse datasets, the essential heterogeneity will be accomplished.

- Random features method: Instead of exhaustively considering all features at each node, A portion of the input features is randomly chosen within each tree of the random forest. From this subset the feature exhibiting the greatest information gain is selected to split the node. The number of features in this subset is smaller than the the overall count of primary features. Every tree within the random forest grows to its maximum capacity without pruning, following the CART algorithm. At each node, the Bremen utilizes a

fixed number of attributes, specifically ⌊log_ 2 ⁡ M ⌋ + 1, where M depicts the complete count of input attributes. (Rafique et al. 2019).

- Definition of arbitrary forest: arbitrary forest is a categorizer of a set comprising of choice forest classifications. Every categorizer for every input instance is $h(x, \theta_k)$, where x is an input instance and θ_k is the instruction set for the k^{th} tree. θ are mutually independent, each following the same distribution. For every individual sample, x, each tree generates a prediction. Eventually, the category that receives the highest number of votes from the trees for input x is chosen as the final prediction for sample x. This procedure is referred to as a stochastic forest (Jasim et al., 2022). The stochastic forest methodology has the potential to enhance the forecasting precision relative to the classifier tree. In the specific tree, instability arises from minor variations in the training set, thereby impacting the prediction accuracy within the experimental sample. However, the aggregation process employed by the random forest algorithm adjusts to changes and mitigates instability.

Before utilizing the arbitrary forest, there are three key variables that need to be adjusted. These variables include the node size, the number of trees, and the number of sampled features. Random forest classification is applicable to both regression and grouping problems. The algorithm consists of multiple decision trees, with each tree using a subset of data from the training dataset. One-third of the data samples are excluded as out-of-bag samples for testing. To enhance dataset diversity, additional random samples are included during the prediction process, drawn from the feature pool. This inclusion of random samples helps reduce the correlation among the decision trees. The prediction outcome varies based on the problem type. For regression or recursive cases, the decision trees are averaged, while for categorical predictions, the majority vote is employed to determine the most common group variable. Finally, the samples excluded from the training set (out-of-bag samples) are utilized for validation and finalizing the predictions.

The decision tree consists of two types of attributes: categorical attributes (also known as batch traits) and numerical attributes (referred to as real traits). Categorical attributes

can take on two or more discrete values or symbolic traits. On the other hand, numerical attributes are associated with a range of real numbers as their values .(Breiman 2001)

The benefits of the decision tree include the following:

1. Complex global decision regions (particularly in expansive environments) can be approximated by combining elementary local decision regions at various levels of the tree.

2. Unlike common one-step classifiers in which each data sample is evaluated on all categories, in a foliage categorizer, an instance is exclusively examined on particular divisions of categories and calculations are eliminated.

3. Single-stage classifiers typically utilize a subset of attributes to differentiate between categories, often selected based on an optimal global criterion. In contrast, tree classifiers offer flexibility in selecting various partitions of characteristics at varying internal nodes of the tree. This allows for the optimal differentiation between categories within each node. Such flexibility can lead to performance enhancements compared to single-stage classifiers.

4. In multiplicity analysis with a large number of attributes and categories, it is usually necessary to estimate large-dimensional distributions or specific variables pertaining to the distribution categories could serve as the starting probabilities within a limited set of training data. there is a high-dimensional problem that can be solved in the classifier tree by using fewer attributes in each internal node without drastically reducing effectiveness.

Variables such as; the quantity of basic learners and the criterion for evaluating the power of dispersal of traits in random forest can be optimized.

3.5.3 Support vector machine

SVM belong to the category of supervised machine learning algorithms. It belongs to the class of discriminative classifiers assign by a discriminative hyperplane. Given a set of data points {(x _ 1, c _ 1), (x _ 2, c _ 2), ..., (x _ n , c _ n)}, the objective is to Partition them into two distinct categories, where c_i ∈ {-1, 1}. Each x_i represents a p-dimensional vector of features. (Elmrabit et al. 2020). LC techniques endeavor to segregate data by constructing a hyperplane (which corresponds to a linear equation). Among these techniques, the SVM classification method, which falls under the category of linear classification methods, determines the optimal hyperplane that maximally separates the data pertaining to the two classes. To enhance comprehension, Figure 3.9 visually illustrates a dataset comprising two classes, wherein SVM identifies the most suitable hyperplane to separate them.

The generation of the separating surface cloud occurs as follows. A comprehensive depiction of the process by which the separating surface is constructed using the support vector machine is presented in Figure 3.12.

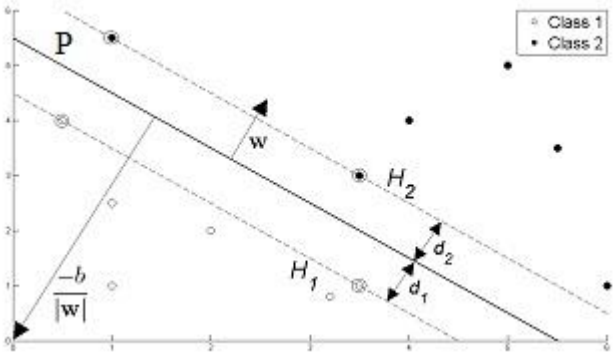


Figure 3.12 How to build a separating surface cloud amongst two classes of data in a two-dimensional space

First consider a convex shell encircling the data points of each class. In Figure 3.12, a convex shell is drawn around points corresponding to class-1 and points corresponding to class +1. The line P is the line that shows the nearest proximity between two convex

shapes shells. h, which is actually the separator superficial, represents a linear connection halves P in the middle and forms a right angle with it.

b is the width of the origin for the supersurface with the maximum separator boundary. If b is ignored, the solution comprises solely the superficials that traverse the origin. The vertical separation between the cloud surface and the origin is determined by dividing the absolute magnitude of parameter b by w.

So, The primary concept is to select an appropriate delimiter. delimiter means a superficial that has the greatest distance from neighboring points on both floors. This answer actually has the most boundaries involving points associated with distinct levels and can be enclosed by two parallel superflocks that intersecting at least one of these points from the two floors.

$$w.x - b = 1 \tag{3.22}$$

$$w.x - b = -1 \tag{3.23}$$

It is noteworthy that if the educational the data can be linearly separated, two boundary surface clouds can be chosen so the absence of data points between them allows for the maximization of the distance between the two parallel surfaces. Using geometric theorems, the separation gap between these two supersurfaces is 2/|w|, So it should | w | Minimized. Data points should also be avoided within the boundary, for which a mathematical constraint is added to the precise meaning. For every i, by imposing the following restrictions, it is guaranteed that no point is placed on the boundary:

$$w.x_i - b \geq 1 \tag{3.24}$$

$$w.x_i - b \leq -1 \tag{3.25}$$

The SVM algorithm described in the previous paragraph is linear and cannot be used for nonlinear data. But they can be generalized to be usable for data that are not linearly

separable. This algorithm is called nonlinear SVM algorithm and is able to create nonlinear superfaces in the input space.

The nonlinear SVM consists of two primary stages. first stages, with the help of a nonlinear mapping, the initial data set is transformed into a space with higher dimensions. There are several nonlinear maps that can be used. After converting the data to a new space with higher dimensions, the algorithm in the second step looks for a linear separator screen cloud in the new space. At this stage, as before, we will face a quadratic optimization problem. the highest gap superset found in data space is like the non-linear separator superspace in the original data. To estimate the class of an experimental sample, the multiplication between each support vector and that sample must be calculated. In practice, to solve a quadratic optimization problem related to linear SVM, it often happens that instances of training data are expressed as multiplication by $\emptyset(X_i).\emptyset(X_j)$, where $\emptyset(X)$is a nonlinear conversion function (Enigo et al. 2020).

Instead of calculating multiplication on newly converted data, a function called the kernel function can be used for the original data. Thus, whenever the calculation of the learning algorithm requires the calculation of $\emptyset(X_i).\emptyset(X_j)$, Because the kernel function is executed on the original data and the magnitude of this information is lesser in comparison to the newly converted data, the calculations will be performed faster. Then, comparable to the method described in the preceding segment, we look for a page with a maximum margin. There exist numerous kernel functions, including the linear one, polynomial kernel, Gaussian radial kernel for SVM, each with its own characteristics, and there is no hypothetical procedure for selecting a better kernel function. In the equation 3.26 the linear kernel was shown and in the equation 3.27 the Gaussian radial.

$$K(X_i, X_j) = X_i^T * X_j \tag{3.26}$$

$$K(X_i, X_j) = \exp\left(-\frac{||X_i - X_j||^2}{2\sigma^2}\right) \tag{3.27}$$

In SVM, Factors like kernel function, penalty coefficient and kernel settings will be optimized.

3.5 Criteria for Evaluating the Proposed Method

The CM will utilized to evaluate and calculate the effectiveness criteria in the proposed approach. To assess/evaluate the method, first the data (samples) are separated into two categories, normal and attack. Accordingly attacks in the attack category and Normal class data are considered as non-attack or normal examples.

The confusion matrix is shown in Table 3.2.

Table 3.2 Confusion Matrix

	Predicted Class (normal)		
Actual Class (normal)		Class=Yes	Class=No
	Class=Yes	True Positive (TP)	False Negative (FN)
	Class=No	False Positive (FP)	True Negatice (TN)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+TN} \tag{3.28}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{3.29}$$

$$\text{Precision} = \frac{TP}{TP+FP} \tag{3.30}$$

In these formulas and the confusion matrix, each of the concepts is defined as follows:
True Positive (TP): Samples whose main category was positive and correctly placed in the positive category.
False Negative (FN): Samples whose main category was positive but incorrectly categorized as negative.
False Positive (FP): Samples whose main category was negative but incorrectly placed in the positive category.
True Negative (TN): Specimens whose main category was negative and correctly classified as negative.

3.6 CNN

CNN is a specialized type of (ANN) is extensively employed in the realm of artificial intelligence. However, what sets CNN apart is its ability to handle complex data patterns by leveraging convolutional operations. This network can be extended by adding more layers, overcoming challenges associated with increased depth, earning them the classification of deep neural networks (Hu et al., 2021).

3.7 Different Types of CNN Models

LeNet: When talking about convolutional neural networks (CNN), the first convolutional architecture that used the back propagation process, LeNet-5 architecture, is definitely mentioned; In fact, it can be said that convolutional neural network and deep learning started with the introduction of LeNet-5 architecture. Before the emergence of LeNet-5 architecture, character recognition was done using manual feature engineering (Feature engineering by hand) and (Machine Learning) to learn the classification of these features. LeNet-5 architecture eliminates manual feature engineering work; because the network itself automatically extracts the best features from the raw input images.

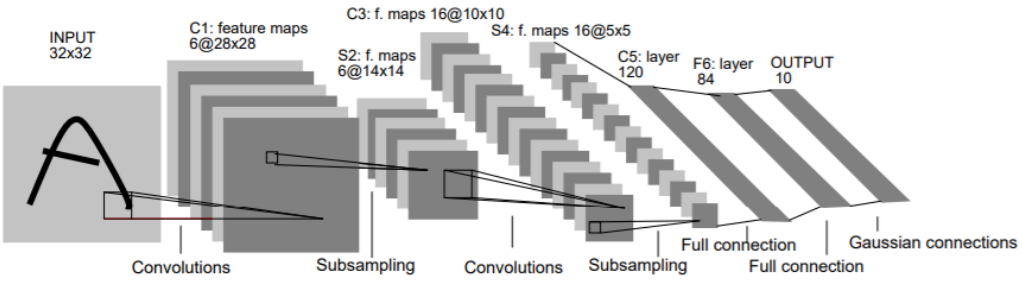


Figure 3.13 LeNet Model

EfficientNet: With scale down the network needs to be trained sooner and with scale up, the network needs training for a longer period of time. For example, if we want our network to be trained faster and a slight decrease in the accuracy of the network results

is not a very important issue, this method can be used. Therefore, Efficient-net is a way to get the best amount for scaling up according to the existing conditions.

ResNet: The ResNet article is one of the important and key articles published in the field of deep learning, which has been referred to more than 100,000 times so far, which is the most referenced article in the world of DL so far, and it is unlikely that it will get its title from give a hand We know that the deeper the neural networks get, the more complicated their learning process becomes, and this article tries to simplify the learning process for optimization algorithms and increase the depth of the neural network as much as it wants. As an example, a network with a depth of 1200 layers is also introduced in the last part of the article. In those years, deep networks had about 20 layers (VGG network with 19 layers and GoogLeNet with 22 layers), and designing and training a network with 1200 layers is considered a great progress, which of course has its own challenges.

VGG: Google Net, also known as the Inception Network emerged as the victor of the ILSVRC 2014 competition, attaining a top-5 error rate of 6.67%percent, comparable to human-level performance. Developed by Google, this model is an enhanced version of the original LeNet design, incorporating the concept of inception modules. GoogLeNet, a variant of the Inception Network, is a deep convolutional neural network comprising 22 layers.Presently, GoogLeNet finds application in diverse computer vision tasks such as face detection, identification, adversarial training, and more. Its versatility makes it a valuable tool for various computer vision applications.

3.8 Chapter Summary

In this chapter, we have tried to describe the details of the proposed method. Accordingly, at first, the general process of the proposed method was described. Then the data set used in this study was examined. Then the feature selection procedure was described and then the classification algorithms were introduced and finally the evaluation criteria of the proposed method were described.

4. RESULTS AND DISCUSSION

4.1 Machine Learning

Within this section, We have developed a methodology that utilizes decision tree classification algorithms in machine learning to detect intrusions in Internet networks. also, we integrate the (GWO) to decrease the dimensionality of features and conduct feature selection. To effectively identify Internet attacks, ur approach involves the utilization of random forest, classification and regression tree algorithms. The evaluation of our proposed methodology is implemented using the NSL-KDD dataset. This section delivers a thorough depiction of our proposed approach, commencing with an introduction to the essential implementation tools, followed by a detailed discussion of the procedural aspects of the methodology.

4.2 Feature Selection with Grey Wolf Optimization

In this section, the feature selection process employed the grey wolf optimization (GWO) algorithm, utilizing the wrapper technique. Drawing inspiration from the natural behaviors of grey wolves, which are top predators and live in collaborative packs, the GWO algorithm aims to approach an optimal solution similar to hunting prey. These leader wolves guide the remaining wolves, known as omega wolves, to update their positions (Devi et al. 2017). The wolves' fitness values are recalculated after position updates. If any wolf has moved to a better position, this iterative process continues until termination conditions are met. The steps of the GWO algorithm can be summarized as follows:

Initialization: Generate the initial population of the wolf pack.

While the termination conditions are not met, perform the following pathway repeatedly:

2.1. Compute the fitness values of all wolves within the pack.

2.2. Identify the delta, alpha, beta and wolves by considering their respective fitness values.

2.3. Adjust the positions of all wolves in the pack based on the positions of the delta, alpha, beta wolves.

2.4. Return to Step 2.

Choose the alpha as the ultimate solution for the Grey Wolf Optimizer algorithm.

During feature selection process, the performance of the model generated by the agents is evaluated using Equation 1 as the objective function. The objective is to choose non-repetitive and informative features while minimizing the classification error rate.

$$\text{Performance}(\text{Model}) = \text{W}_1 * Error(\text{Model}) + \text{W}_2 * \frac{\text{SF}_\text{i}}{\text{TF}} \quad (4.1)$$

Equation 1 combines the error of the random forest classifier (Error(Model)) and the ratio of selected features (SF_i) to the total number of features (TF). A smaller value of (SF_i)/TF indicates a better agent. The weights W_1 and W_2, assigned values of 0.8 and 0.2 respectively, determine the significance of the error and feature selection in the overall performance assessment.

Finally, $\text{W}_1 = 0.8$ and$\text{W}_2 = 0.2$.

4.3 Classification (Decision Tree & Random Forest)

In this section, the Classification and Regression Tree (CART) algorithm was employed. Specifically, a decision tree was used as a classification algorithm, where

samples are categorized as tree progresses traverse from the root to the leaf nodes in a downward direction. Each n-leaf or internal is associated with a feature that poses Inquiry regarding the input selection. Based on feature's value to determine the subsequent step or branch to follow. In a decision tree, the leaf nodes symbolize the ultimate classes. For instance, as sample traverses tree, It progresses from the root to a leaf node in order to ascertain its associated class. Ultimately, it ends up at a leaf node, which signifies the assigned class. (Rafique et al.2019).

The name "decision tree" is attributed to this method because it resembles the process of decision-making when classifying an input sample. Decision trees offer a visual representation of the relationships within a dataset and are widely employed for tasks Incorporating both classification and prediction aspects.

Attributes can be categorized into two types: CA and RA. CA refer to those can take on two or more distinct values, often represented as symbolic. On other hand, real attributes are characterized by values that are obtained from numbers.

- The primary objectives of using decision trees in classification are as follows:
- Accurate classification of input data.
- Building a model that can effectively predict the classes of new data based on training data.
- Easy expandability of the decision tree when new training data is added.
- Striving for simplicity in the resulting tree structure.

The design process of a decision tree involves the following steps:

- Selection of the appropriate decision tree algorithm.
- Identification of the relevant features to be used for decision-making at each internal node.
- Determination of the decision-making rule or strategy to be employed at each internal node.
- Applying various rules or strategies can lead to the creation of distinct decision trees.

It possesses numerous attractive qualities. Characteristics from an algorithmic perspective, including:

- The capability to seamlessly handle both regression and classification tasks.
- Relatively quick training and forecast times in comparison to other algorithms.
- Dependence on only a small number of tuning parameters for optimal performance.
- Capability to estimate the generalization error, providing insights into model performance.
- Effectiveness in solving high-dimensional problems, making them suitable for datasets with many features.
- Ease of implementation within a parallel computing framework, allowing for efficient computation on parallel architectures.

It’s also attractive from a statistical standpoint for the following reasons:

- Measurement importance: Random forests provide a measure of the importance of each feature in the classification or regression process, allowing for feature selection and interpretation of the model.
- Weighting of different classes: Random forests handle imbalanced datasets by assigning appropriate weights to different classes, ensuring fair representation and accurate predictions.
- Handling of missing values: Random forests have the ability to handle missing values effectively, reducing the need for imputation or data preprocessing steps.
- Visualization capabilities: Random forests can be visualized, offering insights into the decision-making process and the relationships between features.
- Outlier detection: Random forests are robust to outliers, minimizing their impact on the model's performance and reducing the risk of overfitting to extreme values.

Random forests function by combining multiple classifiers, with decision trees serving as the baseline classifiers. The (RF) model functions by averaging the outputs of each individual decision tree present in random forests ensemble. Numerous decision trees are generated in a random forest. When a new sample is presented, the computations of each decision tree in the random forest are assessed. The conclusive outcome for the provided example is ascertained through a voting mechanism that depends on the produced results.

The identification process involved a comprehensive examination of prior studies, with many of these datasets having been utilized in prior studies. Elaboration on the information pertaining to the data is furnished in the following segment. Following data collection, non-numerical data were converted into acceptable formats using Excel for classification purposes. Subsequently, the data underwent division into distinct training and test sets.

The purpose of utilizing the training data was to construct classification models, whereas the test data were employed for performance evaluation and the computation of efficiency metrics.

Dataset: Within this particular section, the NSL KDD dataset, originally presented by Khraisat et al. (2019), was employed. The NSL KDD dataset is a modified iteration of the KDD CUP99 dataset, comprising a total of 41 distinct features. It encompasses 22 different attack types, which are primarily categorized (DOS), (U2R), (R2L), and probe groups. Henceforth, the dataset encompasses a total of five distinct classes, with four classes representing different attack types and the fifth class representing normal or healthy data.

For the purpose of this paper, the focus was initially on two classes: healthy data and attack data. Thus, the classification task in this step involved a two-class classification problem. Following this, the efficiency of each attack was analyzed and evaluated.

4.4 Evaluation Criteria

The evaluation of classification efficiency in this paper involved the use of evaluation criteria. Specifically, to calculate these criteria, a confusion matrix was employed. The resulting confusion matrix is presented bellow, which was used for evaluation purposes.

Table 4.1 Confusion matrix for evaluating the performance of classification

		Real class of a sample	
	Total samples	Negative	Positive
Outputs of classifiers (estimated class)	Positive	TP	FP
	Negative	FN	TN

4.4.1 Outputs

For the purpose of feature selection, the (GWO) algorithm was employed. The algorithm began with an initial population of 12 wolves, and The maximum number of iterations was defined as 50. The convergence diagram of the GWO algorithm can be seen in Figure 4.1, which demonstrates the progress and convergence of the algorithm over the iterations.

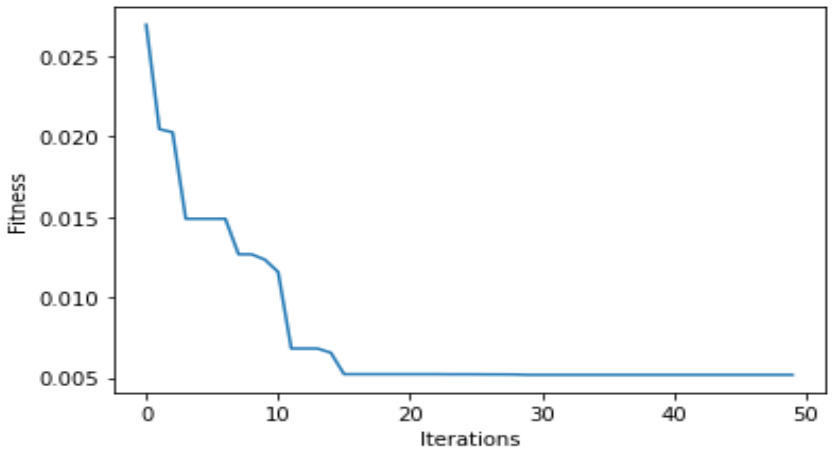


Figure 4.1 The divergence chart of the Grey Wolf Optimizer

Based on shape 1, the dataset was partitioned into a training set comprising 70% of the data and a test set comprising 30% sets using the holdout technique in this study. In this part, the results achieved by implementing the proposed method are presented and evaluated. The outcome both before and after feature selection, are reported and analyzed.

This table presents the outcomes of the suggested approach on the training dataset.

- A total of 100 baseline learners were utilized in the random forest.
- The feature selection process resulted in the selection of 9 features for the analysis.

Table 4.2 The outcomes of the suggested approach on the training dataset

	Before feature selection		After feature selection	
	CART	RF	CART	RF
Precision	99.99	100	99.97	100
Recall	99.97	100	99.94	100
Accuracy	100	100	100	100
F-score	99.99	100	99.97	100

Figure 4.3 Before feature selection

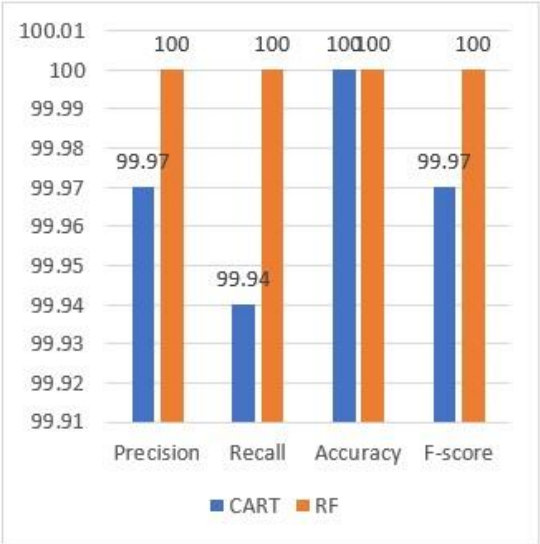


Figure 4.2 After feature selection

Table 4.2 and 4.3 presents the results in percentage values, where higher values (closer to 100) indicate greater efficiency of the model. Based on the observations from Table 2, the subsequent findings can be drawn:

- The arbitrary forest outperformed the CART algorithm in terms of efficiency.
- Both classifiers demonstrated acceptable levels of efficiency.
- It is important to note that these results, obtained from the analysis of proposed method's efficiency cannot be based solely on the results obtained from the training data. Therefore, it is necessary to analyze the outcomes on the evaluation dataset.

Table 4.3 The outcomes of the suggested approach on the testing dataset

	Before feature selection		After feature selection	
	CART	RF	CART	RF
Precision	99.83	99.99	99.80	99.93
Recall	99.80	100	99.87	100
Accuracy	99.87	99.97	99.73	99.87
F-score	99.83	99.98	99.80	99.93

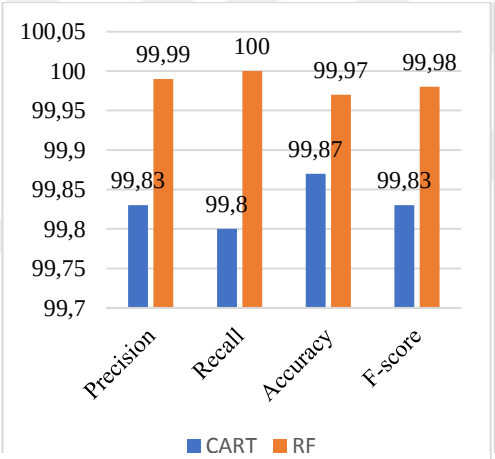


Figure 4.4 Before feature choice

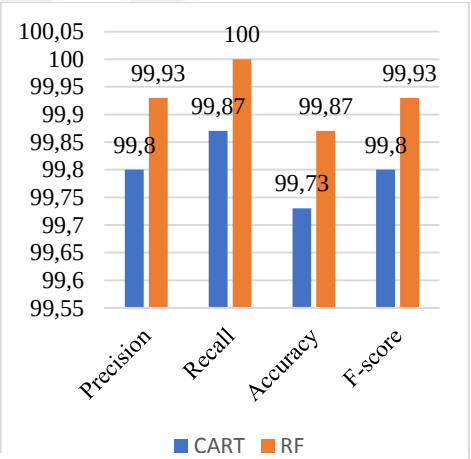


Figure 4.5 After feature choice

The results for each class on multiclass mode are reported below. It is important to highlight that there were five classes in this particular case.

Table 4.4 The results of the suggested approach in multiclass on the test datase

Attack Class	Normal		Dos		Probe		R2L		U2R	
Algorithm / Metric	RF	DT	RF	DT	RF	DT	RF	DT	RF	DT
Accuracy	98.62	98.56	99.76	99.76	99.76	99.69	99.1	99.07	99.81	99.77
Precision	97.76	97.61	99.84	99.84	99.53	99.17	97.11	96.76	95.04	88.62
F1-Score	98.37	98.31	99.64	99.64	98.94	98.65	96.01	95.90	94.26	92.77
Recall	98.98	99.01	99.43	99.43	98.36	98.13	94.94	95.05	93.5	97.32

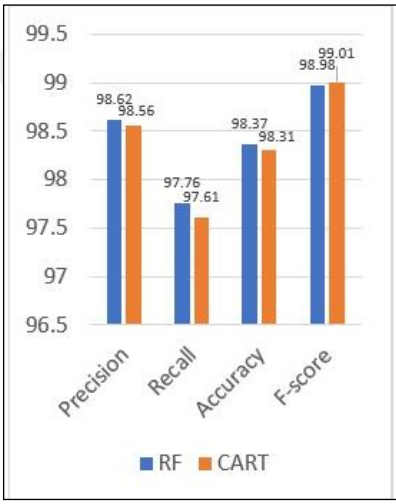


Figure 4.6 Normal attack

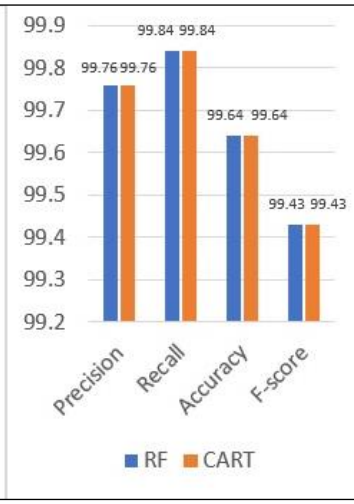


Figure 4.7 Dos attack

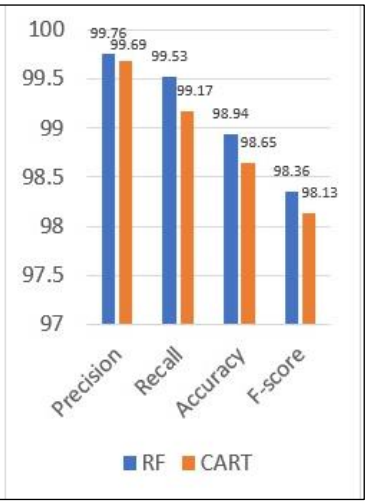


Figure 4.8 Probe attack

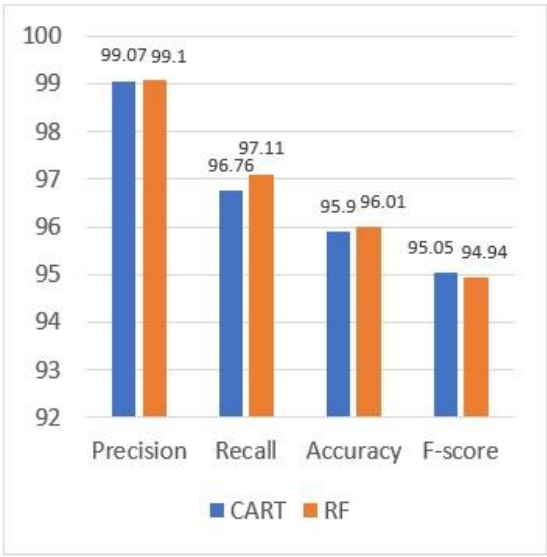


Figure 4.9 RL2 attack

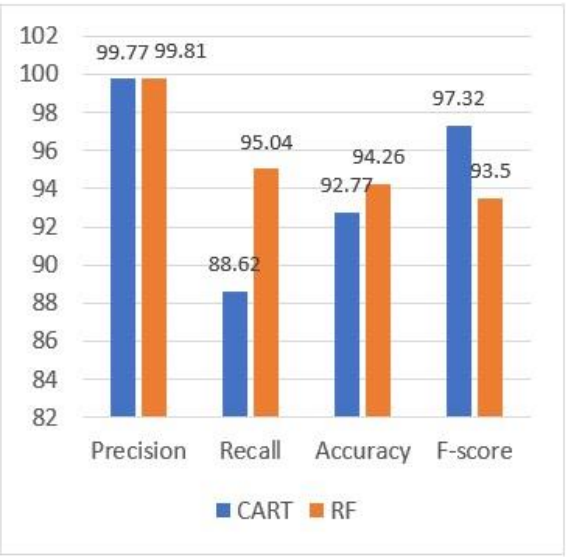


Figure 4.10 UR2 attack

According to the findings derived from the test, it can be inferred that the suggested method demonstrated acceptable efficiency in ID. The subsequent key observations can be established:

- When employed correctly, MLA can exhibit exceptional efficiency in detecting intrusions.
- The arbitrary forest algorithm, as an ensemble learning approach, outperformed the individual CART algorithm in ID.
- While feature choice impacted the effectiveness of the models, it also contributed to reducing their complexity.
- The examination of outcomes on both training and evaluation data indicates that the suggested model did not exhibit over fitting.

4.5 Deep Learning

This section presents a solution that combines meta-heuristic algorithms and machine learning tools, specifically deep learning in combined with traditional learning techniques, such as classification, with the aim of detecting attacks on Internet networks. The approach involves applying deep learning methods for classification and feature extraction to identify intrusions. Furthermore, meta-heuristic algorithms including the grey wolf optimization algorithm, NSGAII (Undominated NSGA II), MOPSO, and Efficient Net are utilized to optimize the parameters of the learning models and perform feature selection.

Dataset: In this section, NSL KDD, DEFCON, CDX, KDD CUP 99, datasets were utilized. NSL KDD dataset is an altered variant of the data, comprises 22 categories of attacks. These malicious activities are primarily classified into DOS, U2R, R2L, and probe groups.

4.5.1 Data preprocessing

Each dataset needs some preprocessing before it can be fed into a machine learning algorithm. Taking NSL-KDD dataset as an example, it has both numerical and categorical features. As far as us aware of, it is impossible or very difficult to use such a dataset without preprocessing. In fact, we need to convert categorical features into numerical. Another issue is that some numerical features are discrete and some are continuous. Moreover, we are going to a feed decision tree algorithm which receive 1D features and a CNN which receives 2D (image) features. Therefore, we have to convert categorical and continuous features into discrete features. To do this, we use binning process to transform continuous features to discrete ones and then employ one-hot encoding to convert the features into numerical space. Figure 2 illustrates the process of numericalization. (Pombeiro et al. 2015).

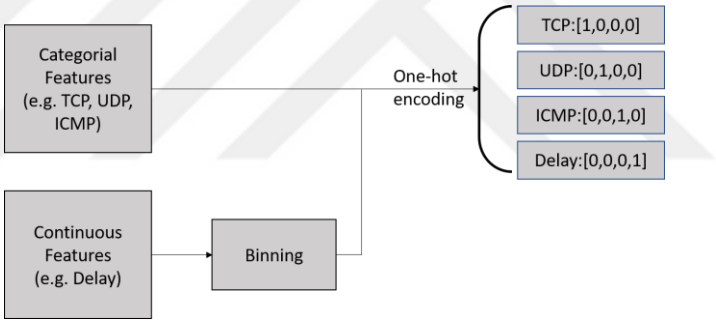


Figure 4.11 Numericalization using binning and one-hot encoding

We also need to make images out of features to feed the CNN algorithm. To do this, we assume every 8-bit data as a pixel of grey scale image.

4.5.2 Pre-training

The primary objective of pre-training stage is to discover the highly efficient features in a dataset. We use metaheuristic algorithms in the loop of a decision tree to optimize accuracy while minimizing the count/number of selected features. Thus, we define a goal function as Eq. (1) and Eq. (2):

$$F = \alpha_1 * (1 - accuracy) + \alpha_2 * SF \tag{4.1}$$

$$\text{SF} = \frac{\text{number of selected features}}{\text{number of all features}} \tag{4.2}$$

Where α_1 and α_2 are two tuning parameters for trading off between precision and the count/number of of chosen characteristics. The output of this stage would be the most effective features in a vector format.

4.5.3 Training

After having the most effective features offered by the combination of Decision Tree and metaheuristic algorithms, we now can train a CNN model. The vital issue to be considered here is to combine 1D (come from pretraining stage) and 2D (come from original dataset) features. We use concatenation concept to join those features.

In machine learning and deep learning, concatenation is a process of joining two or more tensors or vectors along a particular dimension to create a larger tensor or vector. For example, if we have two tensors A and B with shapes (2, 3) and (2, 4) respectively, we can concatenate them along the second dimension to obtain a new tensor with shape (2, 7).

The benefits of concatenation include:

Increased Model Capacity: Enhanced Model Capacity: The capacity of the model can be increased by combining multiple tensors or vectors, resulting in a larger tensor or vector. This expanded size allows to handle more intricate relationships among the inputs and the corresponding outputs, enabling it to learn and represent intricate patterns and dependencies.

Feature Combination: Concatenation can be used to combine multiple features or representations of the input data, which can help capture more informative and meaningful features. This can enhance the model's performance and lead to better generalization.

Information Preservation: Concatenation preserves the information in the original tensors or vectors, as all the elements are included in the resulting tensor or vector. This can be useful when working with time series data, like natural language handling, where preserving the order and context of the inputs is important.

Flexibility: Concatenation can be used in various ways and at different stages of the model. For example, it can be used to concatenate the output of multiple layers of a neural network, or to concatenate the outputs of different models to form an ensemble model.

Figure 3. Depicts the overall procedure of concatenation. This is evident in in Figure 3.

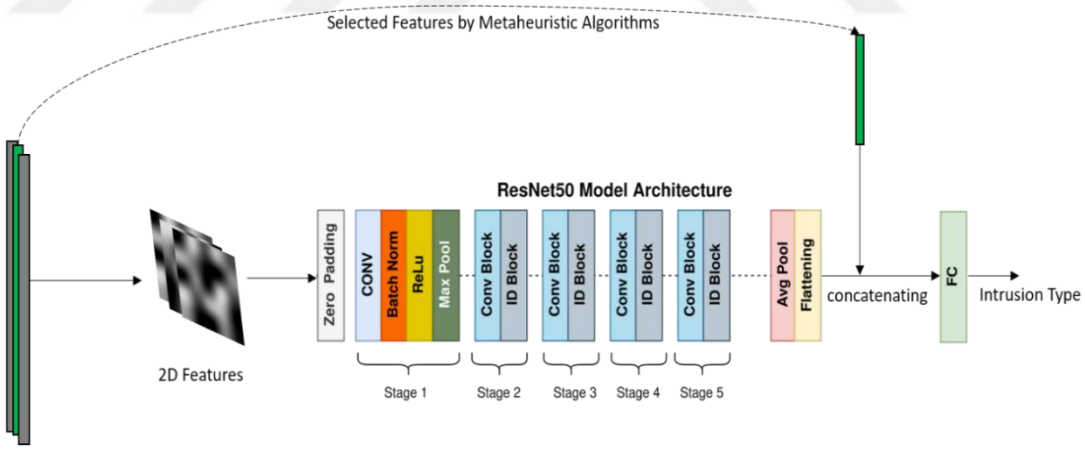


Figure 4.12 Concatenating process. By adding a flattening layer before fully connected layer, it is possible to concatenate features

Where we concatenate features when they are flat or in one dimension. Therefore, we added a flattening layer to CNN models before the fully connected layer which is the point where concatenation has happened.

4.6 KDD-NSL Dataset

4.6.1 Converting non-numeric data to numeric

After loading the data and separating the training and evaluation data and separating the labels from features, we assign a number to the non-numerical data of each feature, according to its classification. The attack type label is also converted into a number based on this:

DoS attack = 1
Probe attack = 2
R2L attack = 3
U2R attack = 4
Normal = 0

To get a better view, the number of data can be plotted according to the type of attack in the diagram:

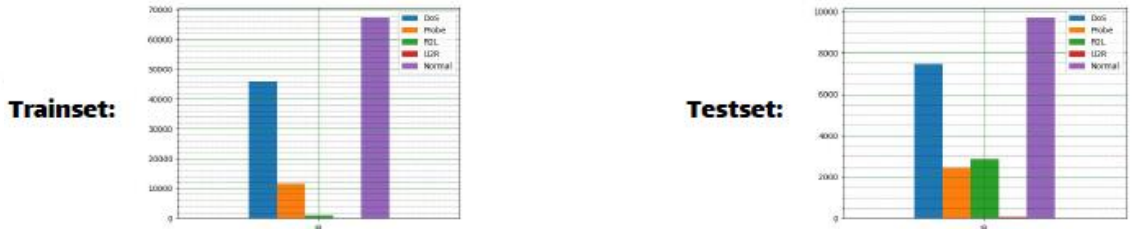


Figure 4.13 Attacks in the diagram

4.6.2 Converting 2D data to images for use in CNN models

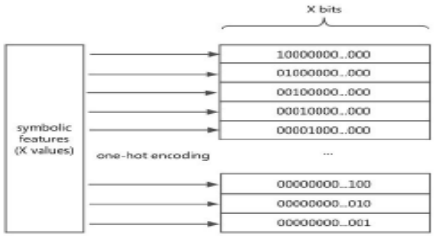
The dataset currently consists of 41 columns and about 125,000 rows, forming a two-dimensional dataset. To test the models .In deep learning, we must first transform the data into an image.

f = ['udp','tcp','icmp']

f[0] = 'udp' = 100

f[1] = 'tcp' = 010

f[2] = 'icmp' = 001



For numerical data; First, we must divide the data into categories with a certain number

Using the equation: $$x_{new} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

We convert the feature to data between (0,1).

Then we classify the data into ten intervals according to the figure below and assign a binary number to each interval as follows:

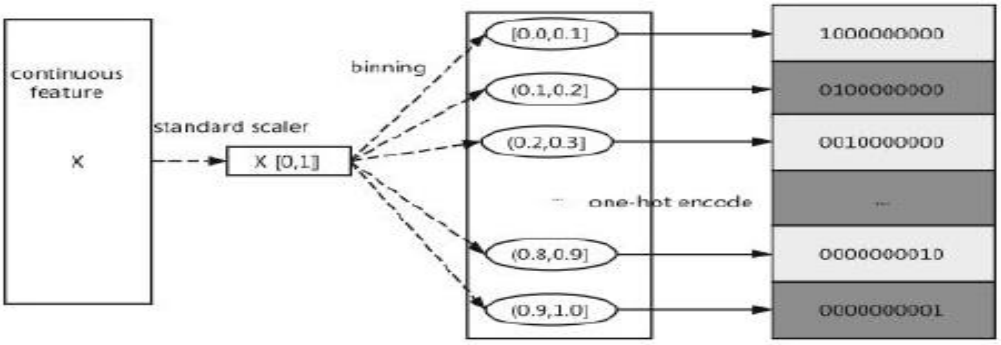
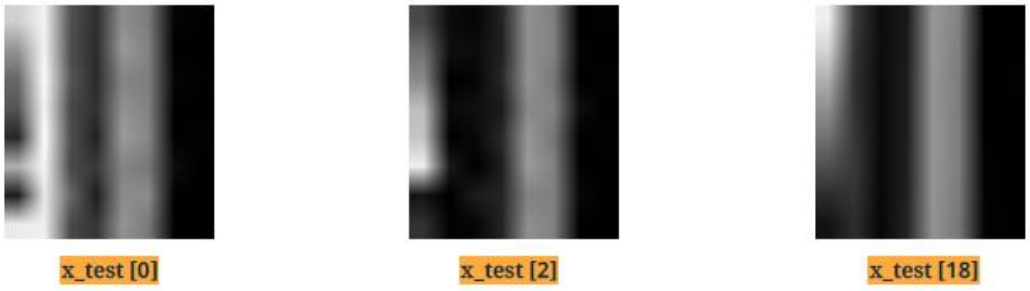


Figure 4.14 Classify the data into ten intervals

Finally, for each data we will have a binary vector with 464 bits. Then we convert every 8 bits into a gray pixel. By adding a sufficient number of zero pixels, the binary vector with 464 bits becomes an 8*8 image

Some examples of saved images after one hot encoding:



Select the model with the highest precision: It was tested and ResNet 50 model's with the best performance was selected.The details related to the time and accuracy of each model on 100 IPAC are as follows:

Table 4.5 Performance report KDD dataset

	train accuracy	test accuracy	total execution time
SimpleCNN	0.9837	0.75	6121.80664563179
LeNet5	0.53491	0.4375	6967.491196632385
ResNet50	0.9899	0.875	37268.8123049736
MyGoogleNet	0.53464	0.4375	61680.567271232605

Table 4.5 displays the test outcomes of four CNN algorithms utilizing the NSL-KDD dataset. SimpleCNN and ResNet50 performed far better than LeNet5 and MyGoogleNet. While SimpleCNN and ResNet50 performed good, but ResNet50 wins the game as its performance is higher than SimpleCNN in many criteria for different dataset.

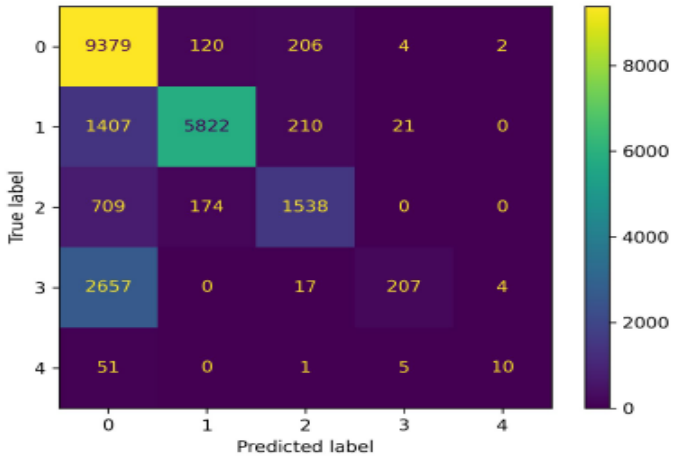


Figure 4.15 Confusion matrix ResNet 50 model

4.6.3 GWO algorithm to enhance the model's performance

Within this approach, we selected features by GWO algorithm individually and after necessary processing such as one hot encoding and... to the end of the feature vector that after passing through the layers conv model are obtained, addition we do.

This method uses the features selected by the gwo algorithm considers a higher coefficient and is likely to have a positive effect have a model performance.

The features chosen by GWO are columns (starting from zero):Selected features: [2, 4, 22, 34, 37, 39]

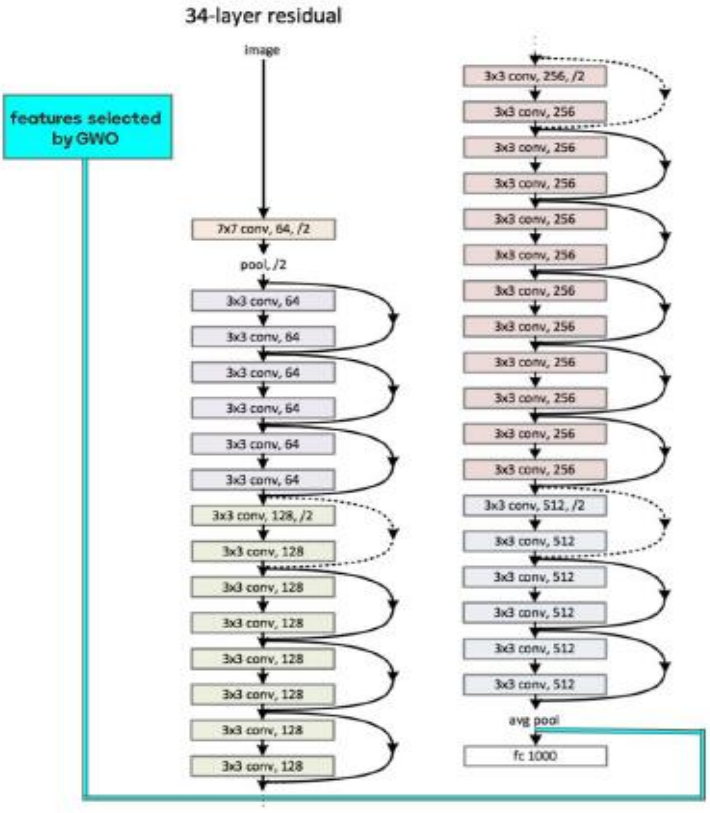


Figure 4.16 The GWO algorithm is utilized to improve the performance

The performance report of this method for 30 rounds is as follows:

Table 4.6 Performance report GWO algorithm

	train accuracy	test accuracy	test f_score	total execution time (Seconds)
ResNet50	0.9718	0.8125	0.74	11599.740952968597
SimpleCNN	0.2195	0.4375	0.2	13611.389083623886
VGG16	0.5345	0.4375	0.43	60672.966148138046

The results in T able 4.6 test results of four C N N algorithms with GW algorithm. ResNet50 and VGG16 performed far better than SimpleCNN . Based on the obtained results, optimal performance is related to ResNet50 with an accuracy of 0.97

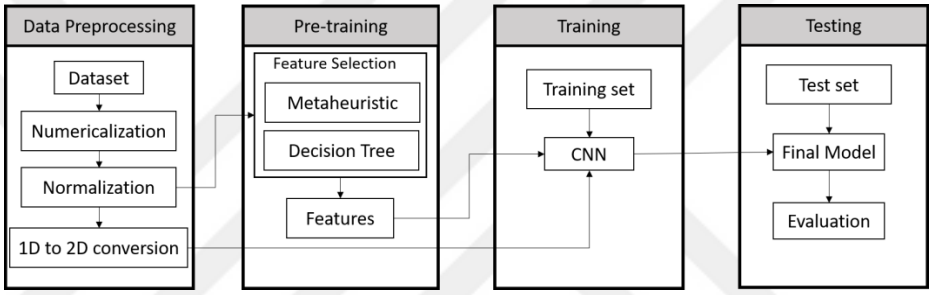


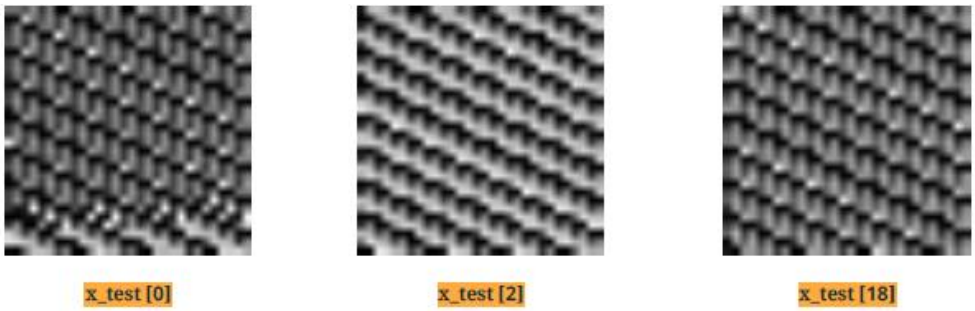
Figure 4.17 Proposed method to combine metaheuristic algorithms and CNN for intrusion detection

4.7 ASNM-CDX- 2000 Dataset

4.7.1 Converting 2D data to images for use in CNN models

After converting the data into binary values using one hot encoding, we will have a binary vector with dimensions of 1*1800 for each line. We convert every ten bits into a gray pixel. By adding enough pixels with zero value, we finally turn each line into a 30x30 picture.

Now the new dataset can be fed to CNN models for training.



Performance test of ResNet 50 model:

On 20 - 30 epoch:
ResNet50, train accuracy:1.0
ResNet50, test accuracy:1.0
ResNet50, test f1 score:0.9887299345388274
ResNet50, test recall:0.9896073903002309
ResNet50, test precision:0.987911577089238
ResNet50, total execution time: 1632.4834389686584 Seconds

-Try GWO method to enhance the model's performance

these features chosen by GWO are columns (starting from zero):
【Selected features: [1, 11, 15, 43, 113, 124, 134,135, 146, 173, 178, 182, 184, 201,214, 230, 238,
245, 247, 251, 256, 271, 351, 370, 373, 374,383,395, 402, 424, 435, 437, 451, 455, 456, 463,
480, 493, 499, 502,560, 579, 586, 611, 619, 646,654, 664, 675, 700, 701, 717, 720,743, 760, 770,771, 786, 787, 797, 805, 815, 824, 830, 843, 844,850, 851, 863]

- The performance report of this method for 30 rounds is as follows:

Table 4.7 Performance report GWO algorithm CDX-DATASET

	train accuracy	test accuracy	test f_score	total execution time(Seconds)
ResNet50	0.9530	0. 75	0.95	595.386506319046
SimpleCNN	0.2445	0.25	0.19	143.5705821514143
VGG16	0.58649	0.25	0.52	3711.785397052765

According to the results presented in Table 4.7, the effectiveness of four CNN algorithms were assessed using the CDX-DATASET dataset. Among the algorithms, ResNet50 and VGG16 demonstrated significantly better performance compared to

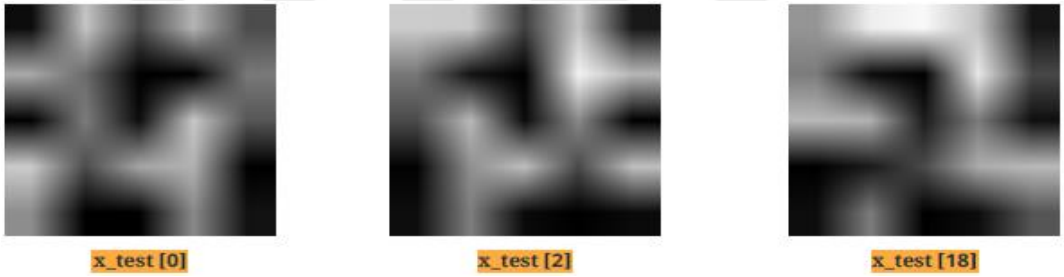
SimpleCNN. Specifically, ResNet50 achieved the highest accuracy of 0.95, indicating superior performance in classifying the dataset.

4.8 DEFCON Dataset

4.8.1 Converting 2D data to images for use in CNN models

After converting the data into binary values using one hot encoding, we will have a binary vector with dimensions of 1*100 for each line. We convert every 4 bits into a gray pixel. By adding enough pixels with zero value, we finally turn each line into a 5*5 picture...

Now the new dataset can be fed to CNN models for training.



Performance test of ResNet 50 model:

DEFCON - unoptimized hyps - 80 epochs:
Best acuuracy on Train set: 0.91875
test_accuracy: 0.625
test_f1_score: 0.4954106519440575
test_recall: 0.4945
test_precision 0.5037964071795785
Total execution time: 3185.694458961487 Seconds

Table 4.8 Performance report GWO algorithm DEFCON DATASET

	train accuracy	test accuracy	test f_score	total execution time (Seconds)
ResNet50	0.482	0. 63	0.47	595.386506319046

SimpleCNN	0.219	0.25	0.19	3466.4848232269287
VGG16	0.534	0.63	0.43	23591.71560382843

Based on the test outcomes presented in T able 4.8, performance of four CNN algorithms was evaluated using the DEFCON DATASET dataset. It was observed that ResNet50 and VGG16 outperformed SimpleCNN by a significant margin. Specifically, VGG16 achieved the highest accuracy of 0.53, indicating superior performance in classifying the dataset. However, it should be noted that Algorithm B (not specified in the paragraph) did not perform well on the selected dataset, suggesting its limited effectiveness for this particular task.

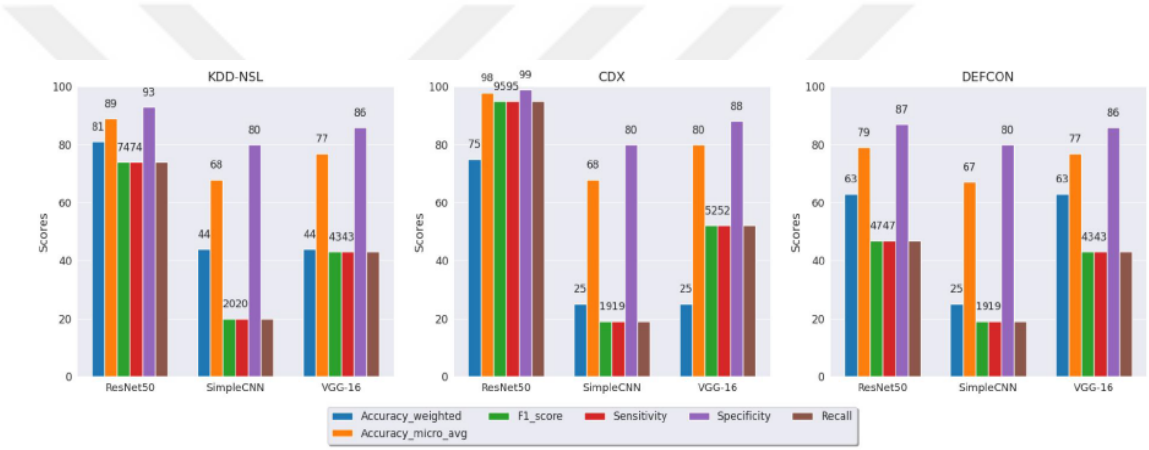


Figure 4.18 The graph of the performance of the models in the GW method for all three data

Figure 4.18 offers a visual analysis of the suggested model's effectiveness in comparison to existing methods and the GW method. The results are depicted for all three datasets. In particular, ResNet50, a specific algorithm within the proposed model, demonstrates superior performance across multiple measures including precision, recall, f-measure, and chi-square values. Additionally, ResNet50 exhibits a shorter processing time, indicating faster and efficient classification of good and bad white wines.

4.9 Repeat Method Using NSGA-II Algorithm

4.9.1 KDD-NSL dataset

Using nsga2 algorithm and arbitrary forest model, the following characteristics are chosen and after retraining the model considering the selected features, the result is as follows:

Model: Random Forest:

NSGA2_Selected_features:[2, 3, 4, 11, 16, 28] # 6 features selected

NSGA2:Subset test accuracy:0.7328779276082328

Execution time: 1594.2539143562317 seconds

Table 4.9 Accuracy of Random Forest model

Attack	Accuracy %
Normal Vs Dos	90
Normal Vs Probe	87
Normal Vs R2L	76
Normal Vs U2R	99
Normal Vs Attacks	72

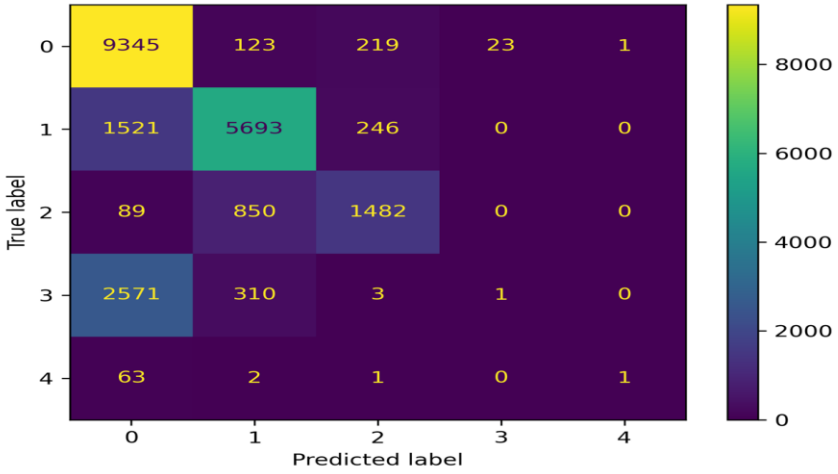


Figure 4.19 Confusion matrix NSGA-II algorithm

-The result of training deep learning models using the mentioned features and methods:

Table 4.10 Performance report NSGA algorithm KDD DATASET

	train accuracy	test accuracy	test f_score	test sensitivity	test specificity	recall	total execution time
ResNet50	0.92	0. 812	0.683	0.683	0.92	0.68	42606
VGG16	0.753	0.437	0.431	0.431	0.857	0.431	66649.13
SimpleCNN	0.219	0.437	0.204	0.204	0.801	0.204	50771.50

The results in Table 4.10 the test four CNN algorithms with KDD DATASET dataset. ResNet50 and VGG16 performed far better than SimpleCNN. Based on obtained results, optimal performance is related to ResNet50 with an accuracy of 0.92.

Table 4.11 Accuracy of ResNet50 model

Attack	Accuracy %
Normal Vs Dos	90
Normal Vs Probe	87
Normal Vs R2L	76
Normal Vs U2R	99
Normal Vs Attacks	72

Table 4.12 Accuracy of SimpleCNN model

Attack	Accuracy %
Normal Vs Dos	51
Normal Vs Probe	52
Normal Vs R2L	51
Normal Vs U2R	53
Normal Vs Attacks	31

Table 4.13 Accuracy of VGG16 model

Attack	Accuracy %
Normal Vs Dos	57
Normal Vs Probe	80
Normal Vs R2L	77
Normal Vs U2R	99
Normal Vs Attacks	43

4.9.2 ASNM-CDX-2009 dataset

Using nsga2 algorithm and arbitrary forest model, the following attributes are chosen and after retraining the model considering the selected features, the result is as follows:

Model: Random Forest:

NSGA2_Selected_features:[1, 5, 8, 34, 42, 45, 47, 78, 102, 103, 129, 131, 132, 144, 147, 149, 154, 164, 171, 176, 181, 197, 226, 227, 237, 238, 242, 251, 252, 266, 270, 277, 278, 284, 300, 332, 333, 341, 362, 365, 397, 401, 406, 417, 418, 432, 433, 446, 448, 451, 469, 496, 501, 508, 521, 542, 549, 551, 556, 558, 570, 576, 587, 591, 601, 604, 606, 616, 617, 630, 632, 647, 676, 690, 696, 730, 734, 746, 749, 767, 772, 777, 786, 796, 819, 824, 827, 835, 844, 850] # 90 features selected

NSGA2: Subset test accuracy: 0.9965357967667436

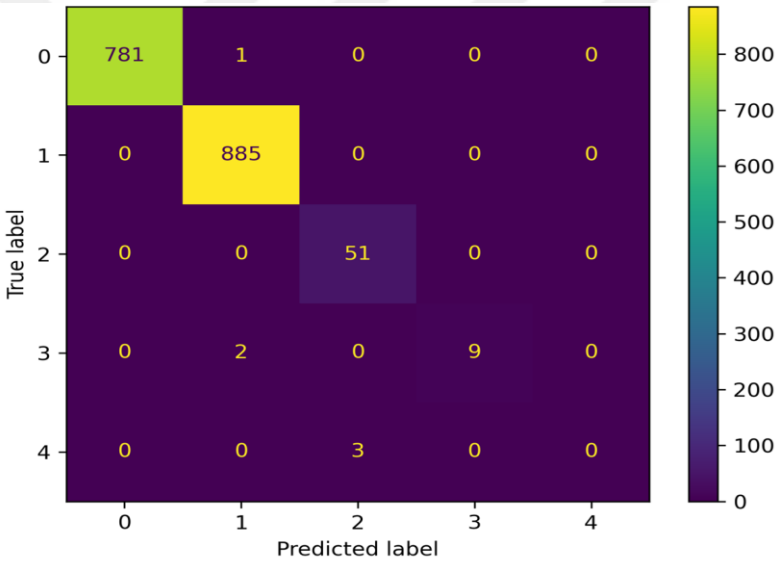


Figure 4.20 Confusion matrix NSGA-II algorithm CDX dataset

The result of training deep learning models using the mentioned features and methods:

Table 4.14 Performance report NSGA algorithm CDX dataset

	train accuracy	test accuracy	test f_score	test sensitivity	test specificity	recall
ResNet50	0.873	0. 75	0.838	0.838	0.95	0.838
VGG16	0.724	0.5	0.722	0.722	0.93	0.722
SimpleCNN	0.246	0.5	0.162	0.162	0.790	0.162

The test results of four CNN algorithms with the KDD DATASET dataset are presented in Table 4.14. Among the algorithms evaluated, ResNet50 and VGG16 outperformed SimpleCNN. Based on the obtained results, ResNet50 achieved the highest performance with an accuracy of 0.87. This indicates that ResNet50 demonstrated better classification accuracy in comparison to the other algorithms when applied to the KDD DATASET dataset**.**

4.9.3 DEFCON dataset

Using nsga2 algorithm and arbitrary forest model, the following attributes are chosen and after retraining the model considering the selected features, the result is as follows:

Model: Random Forest:

NSGA2_Selected_features:[9]

NSGA2:Subset test accuracy:0.5423333333333333

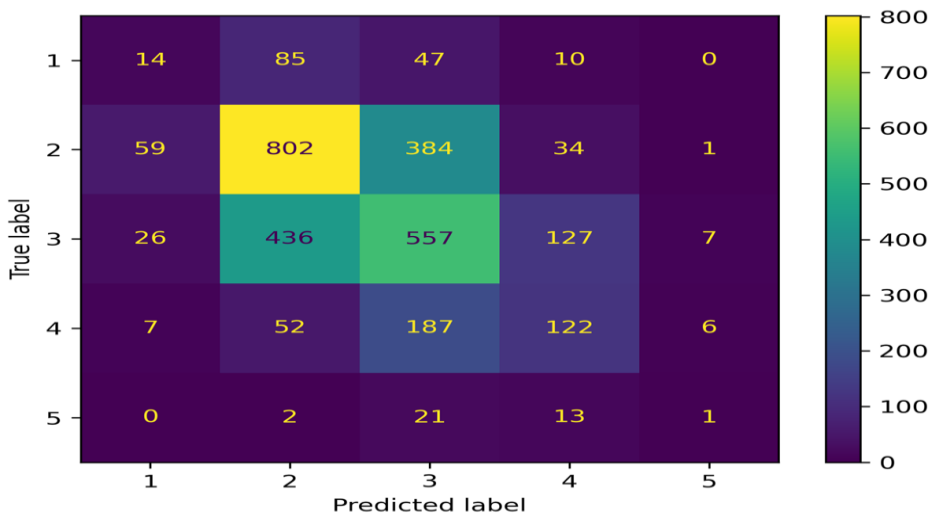


Figure 4.21 Confusion matrix NSGA-II algorithm

The result of training deep learning models using the mentioned features and methods:

Table 4.15 Performance report NSGA-II algorithm DEFCON dataset

	train accuracy	test accuracy	test f_score	test sensitivity	test specificity	recall
ResNet50	0.696	0. 25	0.386	0.286	0.846	0.426
VGG16	0.426	0.625	0.426	0.426	0.857	0.426
SimpleCNN	0.223	0.625	0.212	0.212	0.803	0.213

The test results of four CNN algorithms with the DEFCON DATASET are presented in Table 4.15. Among the algorithms evaluated, ResNet50 and VGG16 showed superior performance compared to SimpleCNN. According to the obtained results, ResNet50 achieved the highest performance with an accuracy of 0.69. This indicates that ResNet50 outperformed the alternative algorithms in relation to classification precision when applied to the DEFCON DATASET.

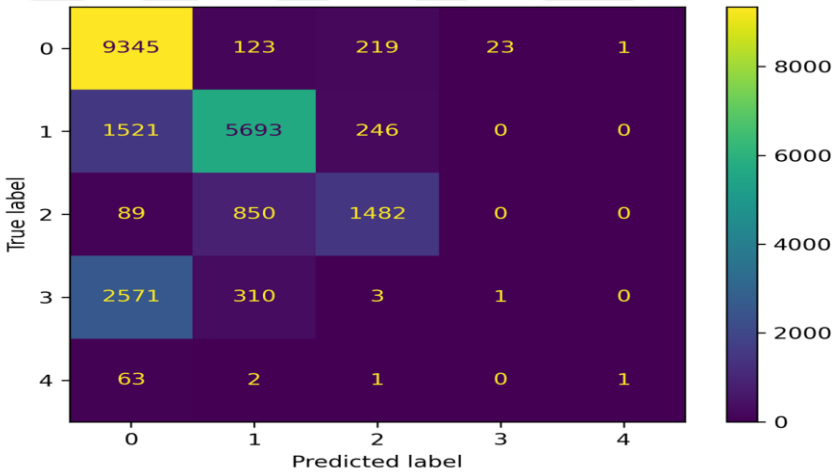


Figure 4.22 The graph of the performance of the models in DEFCON for all three datasets

Fig 4.22 illustrates the effectiveness of suggested model contrasted to the existing methods over the NSGA-II algorithm DEFCON. The presented ResNet50 and VGG16 has achieved higher measures for classifying the good and bad white wines.

4.10 Repeating the Method Using the MOPSO Algorithm

4.10.1 KDD-NSL dataset

Using MOPSO algorithm and arbitrary forest model, the following attributes are chosen are selected and after retraining the model considering the selected features, the result is as follows:

Model: Random Forest:

MOPSO_Selected_features:[1, 2, 5, 10, 11, 16, 20, 21, 29, 34, 37] # 11 features selected

MOPSO:Subset test accuracy:0.7260911994322214

Execution time: 21174.008937120438 seconds

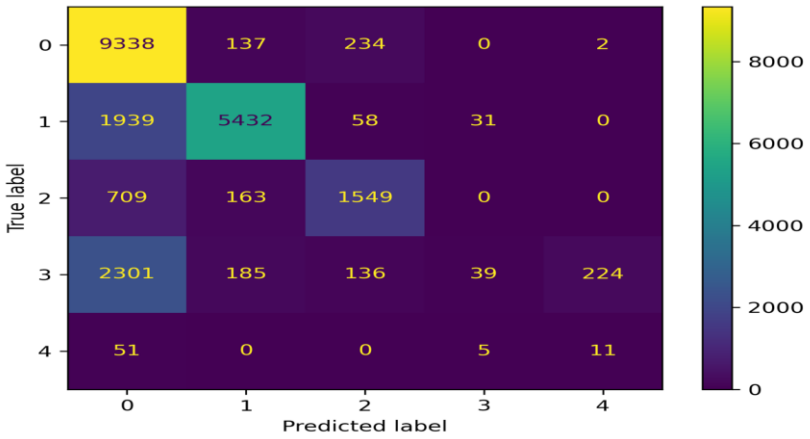


Figure 4.23 Confusion matrix MOPSO

Table 4.16 Accuracy of random forest model

Attack	Accuracy %
Normal Vs Dos	88
Normal Vs Probe	92
Normal Vs R2L	80
Normal Vs U2R	99
Normal Vs Attacks	75

Table 4.17 Performance report NSGA-II algorithm DEFCON dataset

	train accuracy	test accuracy	test f_score	test sensitivity	test specificity	recall
ResNet50	0.812	0. 687	0.687	0.921	0.687	0.42801
VGG16	0.536	0.437	0.430	0.430	0.857	0.430
SimpleCNN	0.22	0.5	0.208	0.208	0.802	0.208

The test results of four CNN algorithms with the DEFCON DATASET are summarized in Table 4.5. Among the algorithms evaluated, ResNet50 and VGG16 exhibited significantly better performance compared to SimpleCNN. According to the obtained results, ResNet50 achieved the highest performance with an accuracy of 0.81. This indicates that ResNet50 outperformed when applied to the DEFCON DATASET.

4.10.2 ASNM-CDX-2009 dataset

Using MOPSO algorithm arbitrary forest model, the following attributes are chosen and after retraining the model considering the selected features, the result is as follows:

Model: Random Forest:

MOPSO_Selected_features: [0, 1, 6, 7, 10, 14, 17, 19, 23, 24, 25, 29, 36, 37, 42, 45, 46, 47, 51, 54, 55, 57, 58, 60, 61, 62, 65, 69, 72, 81, 83, 84, 85, 87, 92, 93, 96, 105, 110, 111, 113, 116, 123, 125, 127, 128, 134, 137, 139, 140, 142, 144, 149, 151, 153, 154, 156, 157, 159, 162, 166, 169, 170, 171, 174, 177, 180, 182, 184, 185, 186, 188, 189, 190, 192, 198, 199, 203, 204, 207, 211, 212, 213, 217, 218, 219, 220, 221, 222, 226, 229, 234, 235, 236, 238, 239, 240, 241, 243, 247, 248, 249, 250, 251, 252, 253, 256, 258, 259, 261, 262, 265, 266, 267, 269, 272, 273, 279, 280, 284, 285, 288, 289, 290, 291, 298, 299, 301, 303, 307, 308, 313, 318, 322, 323, 326, 327, 328, 331, 332, 337, 344, 346, 350, 351, 353, 354, 355, 356, 357, 358, 359, 361, 362, 363, 364, 365, 367, 371, 372, 373, 381, 384, 385, 386, 387, 388, 390, 391, 392, 394, 396, 398, 399, 401, 403, 405, 407, 410, 411, 413, 418, 423, 424, 429, 433, 437, 439, 443, 444, 445, 449, 453, 458, 461, 463, 465, 467, 469, 471, 479, 480, 483, 487, 496, 498, 501, 503, 505, 506, 507, 510, 512, 521, 523, 526, 527, 528, 530, 536, 538, 541, 542, 543, 547, 549, 552, 553, 554, 556, 557, 558, 559, 564, 569, 571, 582, 583, 586, 587, 588, 589, 594, 595, 596, 599, 602, 607, 609, 610, 614, 615, 617, 618, 619, 626, 627, 629, 630, 631, 632, 634, 638, 639, 640, 650, 654, 661, 665, 668, 670, 672, 674, 675, 676, 678, 681, 688, 689, 691, 693, 694, 695, 697, 699, 709, 718, 719, 720, 721, 723, 725, 726, 735, 744, 746, 747, 751, 752, 753, 754, 755, 756, 759, 761, 762, 763, 766, 769, 771, 773, 779, 787, 788, 789, 793, 794, 797, 801, 805, 806, 810, 813, 814, 819, 820, 823, 825, 826, 827, 829, 832, 834, 837, 840, 851, 852, 854, 855, 857, 858, 859, 861, 863, 865, 866, 867] # 347 features selected

MOPSO:Subset test accuracy: 0.995958429561201

Execution time: 4112.666108846664 seconds

Table 4.18 Performance report NSGA-II algorithm ASNM-CDX dataset

	train accuracy	test accuracy	test f_score	test sensitivity	test specificity	recall
ResNet50	0.515	0. 25	0.512	0.512	0.88	0.512
VGG16	0.507	0.25	0.512	0.512	0.88	0.512
SimpleCNN	0.247	0.5	0.22	0.22	0.804	0.22

The test results of four CNN algorithms with the ASNM-CDX DATASET are presented in Table 4.18. Among the evaluated algorithms, ResNet50 and VGG16 demonstrated superior performance compared to SimpleCNN. According to the obtained results, ResNet50 achieved the highest performance with an accuracy of 0.51. This indicates that ResNet50 outperformed alternative algorithms in relation to classification precision when applied to the ASNM-CDX DATASET.

4.10.3 DEFCON dataset

Using MOPSO algorithm and arbitrary forest model, the following attributes are chosen and after retraining the model considering the selected features, the result is as follows:

Model: Random Forest:

MOPSO_Selected_features: [6] # 1

MOPSO:Subset test accuracy:0.465

Execution time: 2546.1442568302155 seconds

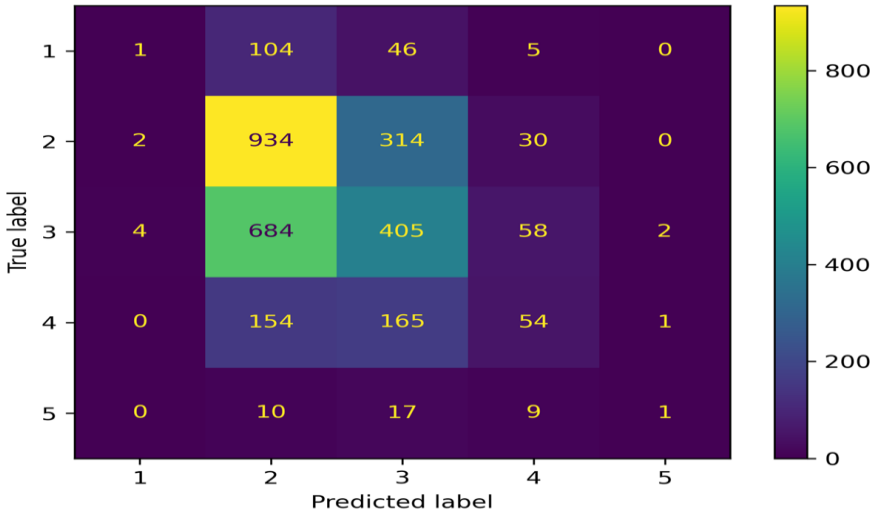


Figure 4.24 Confusion matrix MOPSO algorithm DEFCON dataset

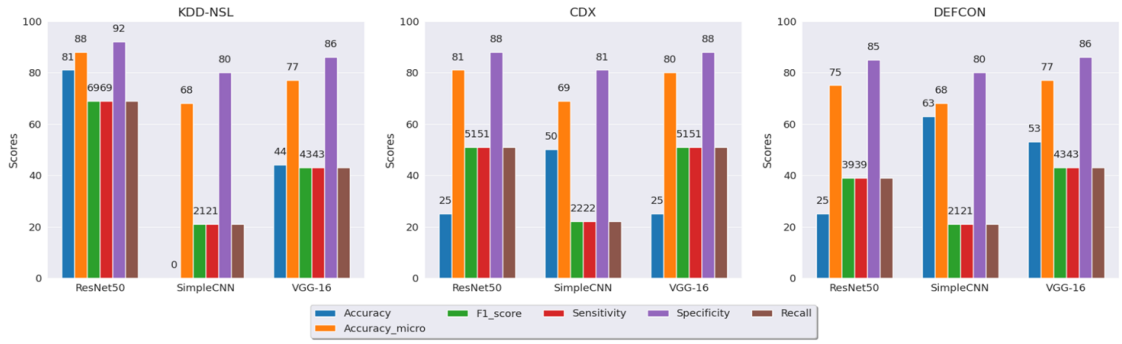


Figure 4.25 The graph of the performance of the models in MOPSO for all three datasets

Fig 4.25 illustrates the effectiveness of suggested model contrasted to the existing methods over the MOPSO for all three datasets. The presented ResNet50 has attained higher measures for classifying the good and bad white wines.

4.11 CNN Architecture and Meta-heuristics Algorithms

We trained three CNN algorithms consisting of EfficientNet, ResNet50, and VGG-16 on different dataset according to the approach proposed in Figure 4.17. Four criteria including accuracy, recall, f1-score and specificity were shown in that figure. On the

other hand Figure 4.25, depicts the same procedure for CDX, DEFCON, NSL-KDD dataset.

Moreover, DEFCON dataset seems to be a challenging task compared to NSL-KDD and CDX. It can be because of the limited number of features that this dataset includes.

Table 4.19 summarizes all the results from different tests.

Table 4.19 Pre-training results for NSL-KDD by Decision Tree

Meta-heuristic algorithm	**No. of features**	**Selected features**	**Selected features (%)**	**Feature Reduction Rate**
GWO	41	[2, 4, 22,11, 34, 37, 39]	17.1	82.9
MOPSO	41	[1, 2, 5, 10, 11, 16, 20, 21, 29, 34, 37]	26.8	73.2
NSGA-II	41	[2, 3, 4, 11, 16, 28]	14.6	85.4

Table 4.20 Pre-training results for CDX by Decision Tree

Meta-heuristic algorithm	**No. of features**	**Selected features**	**Selected features (%)**	**Feature Reduction Rate**
GWO	41	[1,6,8,15,21,…]	24.3	75.7
MOPSO	41	[1, 5, 8, 34, 42, 45,…]	10.3	89.7
NSGA-II	41	[0, 1, 6, 7, 10, 14,…]	39.1	60.9

Table 4.21 Pre-training results for DEFCON by Decision Tree

Meta-heuristic algorithm	**No. of features**	**Selected features**	**Selected features (%)**	**Feature Reduction Rate**
GWO	41	[1,6,9]	30	70
MOPSO	41	[6,8]	20	80
NSGA-II	41	[2,9]	20	80

Table 4.22 The results of tests CNN architecture and meta-heuristics algorithm

Meta-heuristic algorithm	**CNN Architecture**	**Accuracy** [NSL-KDD, DEFCON,CDX]	**Recall** [NSL-KDD, DEFCON,CDX]	**Specificity** [NSL-KDD, DEFCON,CDX]	**F1_Score** [NSL-KDD, DEFCON,CDX]
GWO	EfficientNet	[86.2,**74.0**,78.1]	[**81.1**,66.7,91.3]	[88.6,54.9,90.6]	[**76.8**,**55.8**,93.3]
	ResNet50	[81.3,63.2,75.5]	[**74.0**,47.5**,95.1**]	[89.5,47.3,91.5]	[74.2,47.3,**95.1**]
	VGG-16	[43.8,63.0,25.1]	[40.0,43.0,52.3]	[43.8,47.5,52.5]	[51.2,43.0,52.1]
MOPSO	EfficientNet	[**87.3**,54.0,41.5]	[71.2,**61.2,**62.1]	[**95.2**,**89.1**,91.1]	[75.2,46.7,55.0]
	ResNet50	[81.3, 25.0,25.0]	[68.7,42.5, 51.2]	[92.2,84.7,88.3]	[68.7,38.6,51.2]
	VGG-16	[43.8, 62.5,25.0]	[43.8,42.7,51.2]	[85.8,85.7,88.0]	[43.1,42.7,51.2]
NSGA-II	EfficientNet	[85.2,67.5,**78.8**]	[71.3,40.5,82.1]	[93.1,68.3,95.4]	[67.2, 54.8,87.2]
	ResNet50	[81.3, 25.0,75.2]	[68.3,42.7,83.9]	[92.1,84.7,**95.9**]	[68.32,38.6,83.9]
	VGG-16	[43.8,62.5,50.5]	[43.1,42.7,72.2]	[85.8,85.7,93.1]	[43.1,42.7,72.2]

Using the CNN algorithm demonstrated acceptable efficiency in the field of intrusions. The subsequent findings can be drawn:

1. NSL-KDD dataset using the GWO algorithm: Among the evaluated models, EfficientNet and ResNet50 outperformed VGG16. The best performance was achieved by EfficientNet with an accuracy of 86.2%.

2. DEFCON dataset using the GWO algorithm: The best performance was obtained with EfficientNet, achieving an accuracy of 74.0%.

3. CDX dataset using the GWO algorithm: EfficientNet achieved the highest performance with an accuracy of 78.1%.

4. NSL-KDD dataset using the MOPSO algorithm: EfficientNet and ResNet50 performed better than VGG16. The model with the highest performance was EfficientNet with a precision of 87.3%.

5. DEFCON dataset using the MOPSO algorithm: VGG16 and EfficientNet outperformed ResNet50. The best performance was achieved by VGG16 with an accuracy of 62.5%.

6. CDX dataset using the MOPSO algorithm: EfficientNet exhibited best effectiveness with a precision of 41.5%.

7. NSL-KDD data set using the NSGA-II algorithm: EfficientNet and ResNet50 outperformed VGG16. The model with the highest performance was EfficientNet with a precision of 85.2%.

8. DEFCON dataset using the NSGA-II algorithm: EfficientNet and VGG16 performed better than ResNet50, with EfficientNet achieving best effectiveness with a precision of 67.5%.

9. CDX data set using the NSGA-II algorithm: EfficientNet achieving best effectiveness with a precision of 78.8%.

Overall, the results in Table 4.22 indicate that EfficientNet and ResNet50 consistently outperformed VGG16. EfficientNet showed superior performance in various evaluation criteria for different datasets. Additionally, DEFCON dataset posed a more challenging task compared to NSL-KDD and CDX, possibly due to its limited number of features. Table 4.22 provides a summary of all the results obtained from the different tests.

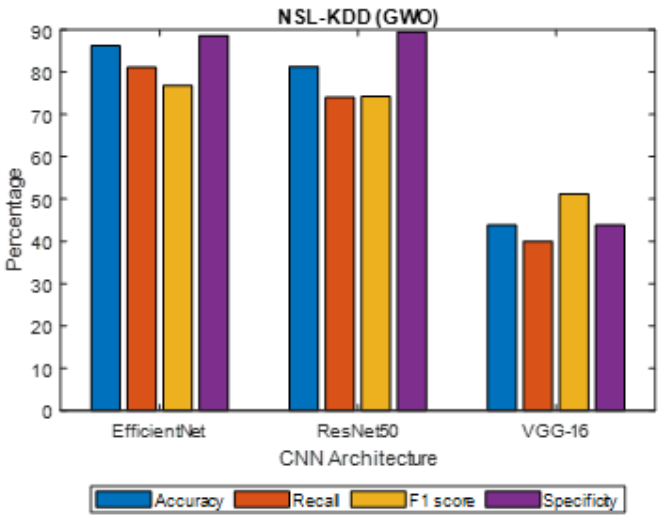


Figure 4.25 CNN algorithms test results for NSL-KDD dataset

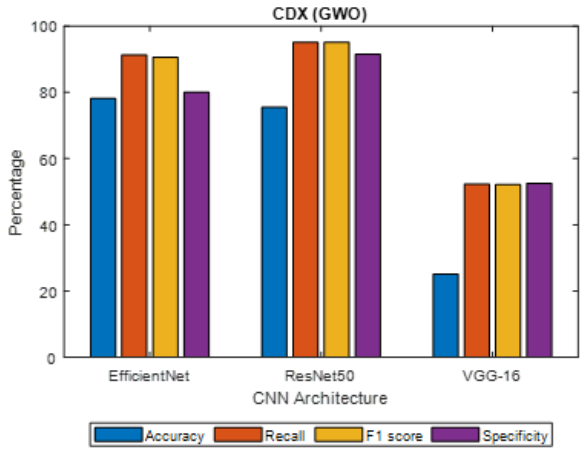


Figure 4.26 CNN algorithms test results for DEFCOND dataset

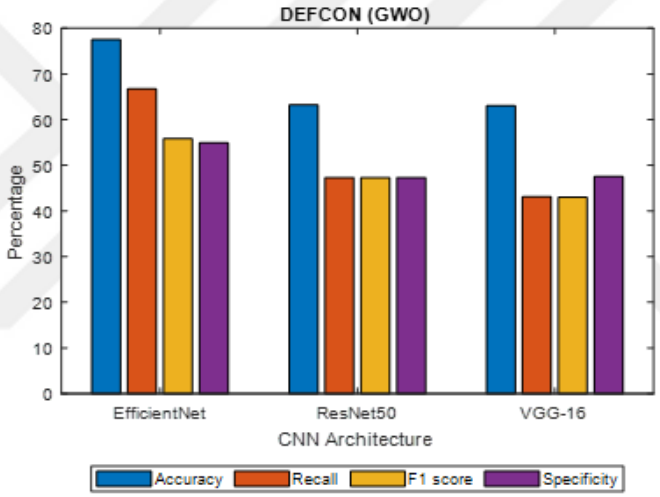


Figure 4.27 CNN algorithms test results for CDX dataset

5. CONCLUSION

Within this study, we introduced an innovative approach that combines metaheuristic-based feature selection with decision trees to identify the most pertinent characteristics in different datasets. These selected features were then utilized as inputs for various CNN architectures, namely ResNet50, VGG16, and EfficientNet. Our proposed feature selection method aimed to enhance the accuracy of the CNN models by identifying strong input features.

The experimental results we obtained demonstrated the effectiveness of our approach across different benchmark datasets and CNN architectures. The combination of the proposed feature selection method and CNN models yielded improved performance in identifying and classifying intrusions in network traffic.

One notable advantage of our approach is its applicability to online and real-time applications. The process of characteristic selection is conducted throughout initial training phase, enabling the method to be seamlessly integrated into live systems.

Overall, our findings highlight the potential of the proposed method for effectively detecting and classifying intrusions in network traffic. The characteristic selection method can be a valuable tool in enhancing effectiveness of intrusion detection systems, particularly when combined with CNN architectures.

Based on the results obtained from the test data, several conclusions can be drawn regarding the effectiveness of the suggested approach for intrusion detection:

- feature selection method implemented in the proposed approach successfully identifies strong input features, resulting in improved accuracy of the CNN model. This underscores the significance of in characteristic selection enhancing the effectiveness of intrusion detection systems.

- The experimental findings demonstrate the potential of the proposed method to effectively identify and classify intrusions in network traffic. This suggests that the combination of metaheuristic-based feature selection and CNN architectures can be a promising approach for breach detection.
- The proposed characteristic selection method is well-suited for online and real-time applications. The process of characteristic selection is conducted throughout initial training phase, the method can be seamlessly integrated into live systems, enabling efficient intrusion detection in real time scenarios.
- It is worth mentioning that the efficiency of the proposed algorithms relies on their correct usage. Proper implementation and configuration of the algorithms are crucial for achieving optimal results in breach discovery tasks.
- characteristics choice not only improves the effectiveness of the models but also decreases their intricacy. By selecting the most relevant features, the models can focus on the essential information and discard irrelevant or redundant features, leading to more streamlined and efficient intrusion detection processes.

The next section will provide a comprehensive conclusion based on the findings and insights obtained throughout the study

5.1 Future Recommendations

Indeed, the proposed method and exploration of metaheuristic algorithms for feature selection have shown high efficiency in intrusion detection. To further advance the research in this field, the subsequent possibilities can be explored for upcoming investigation:

- Comprehensive Datasets: Use more extensive data sets that cover a wide range of intrusion scenarios, including various attack types, network configurations, and real-world network traffic. This will provide a more realistic and diverse data environment for evaluating the proposed method's performance.

- Integration of Multiple MLT: Investigate the combination of multiple MLT, such as combining CNNs with alternative algorithms such as (SVM), Random Forests, or Deep Belief Networks (DBN). Assess the potential improvements in accuracy and efficiency when using hybrid models for intrusion detection.

- Hybrid Classification Methods: Investigate the integration of different classifiers or hybrid classification approaches to enhance the accuracy and robustness of intrusion detection. Combining multiple classifiers, such as ensemble methods or stacking models, can leverage the capabilities of single classification model and enhance overall detection effectiveness.

- Reduction of Training Time: Investigate methods to decreases the training duration of the suggested method, particularly when handling extensive-scale data sets. Techniques like distributed computing, parallel processing, or model compression can be explored to accelerate the learning procedure and improve the scalability of the breach identification system.

- Multi-Modal Methods: Explore the application of multi-modal methods that analyze various aspects of informations, flow-based features, temporal patterns, and behavioral characteristics. Integrating multiple modalities can offer a more comprehension of network intrusions and improve the accuracy of detection.

- Multi-Source Datasets: Collect and utilize multi-source datasets from different data sources, including diverse network environments and attack scenarios. Incorporating data from different sources with various class types can provide a broader perspective on intrusion detection and enable more robust and generalizable models.

By tackling these research concerns avenues, we can progress the field of breach identification and develop more effective and reliable methods for detecting and mitigating security threats in network systems.

REFERENCES

Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020). "A Machine Learning Approach for Intrusion Detection System on NSL-KDD Dataset." In 2020 International Conference on Smart Electronics and Communication (ICOSEC), IEEE, pp. 919-924.

Agrawal, P., Abutarboush H. F., Ganesh, T., and A. Mohamed W., 'Metaheuristic algorithms on feature selection: A survey of one decade of research, Ieee Access, 9, 26766–26791, 2021.

Ahmed, Z. E., Saeed, R. A., Mukherjee, A., and Ghorpade, S. N., "Energy optimization in low-power wide area networks by using heuristic techniques", LPWAN Technologies for IoT and M2M Applications.

Ajitha, D. D., Basil, X. S., David, S., & Kathrine, J. (2019). "Comparative analysis of various Machine Learning Techniques for Intrusion Detection System." In 2019 2nd International Conference on Signal Processing and Communication (ICSPC), IEEE, pp. 348-351.

Alpaydin, E. "Introduction to machine learning", 2nd ed., The MIT Press, 2010.

Alzaqebah, M. et al., 'Hybrid feature selection method based on particle swarm optimization and adaptive local search method', International Journal of Electrical and Computer Engineering, 113, 2414, 2021.

Alzubi, Q. M., Anbar, M., Alqattan, Z. N., Al-Betar, M. A., and Abdullah, R., "Intrusion detection system based on a modified binary grey wolf optimisation," Neural Computing and Applications, pp. 1-13, 2019.

Arivardhini, S., Alamelu L. M., and Deepika, S., "A Hybrid Classifier Approach for Network Intrusion Detection," presented at the 6th International Conference on Advanced Computing & Communication Systems (ICACCS), 2020.

Bachar, A., Makhfi N., and Bannay O. E., "Towards a behavioral network intrusion detection system based on the SVM model," in 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 2020: IEEE, pp. 1-7.

Balachandra, M., & Belavagi, M. C. (2016). "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection." Journal of Computer Science, 89, 117-123.

Breiman, L., "Bagging predictors," Machine learning, vol. 24, pp. 123.140, 1996.

Breiman, L., "Random forests," Machine learning, vol. 45, pp. 5-32, 2001.

Brownlee, J.; 2011, “Clever Algorithms Nature-Inspired Programming Reciptes”; Melbourne.

Chang, Y., Li, W., and Yang, Z., "Network intrusion detection based on random forest and support vector machine," in 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), 2017, vol. 1: IEEE, pp. 635-638.

Chkirbene, Z., Eltanbouly, S., Bashendy, M., AlNaimi, N., and Erbad, A., "Hybrid Machine Learning for Network Anomaly Intrusion Detection," presented at the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, Qatar 2020.

Devi, E. M. and Devi, R. C. x, ‘Feature selection in intrusion detection grey wolf optimizer’, Asian Journal of Research in Social Sciences and Humanities, 7, 3, 671–682, 2017.

Dincalp, U., Güzel, M. S., Sevine O., Bostanci, E., and Askerzade, I., ‘Anomaly based distributed denial of service attack detection and prevention with machine learning’,2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 1–4, 2018.

Doreco, J., Siarry, E., Petrowski, A., and Taillard, E., “Metaheuristics for Hard optimization”`n. Springer-Verlag, 12-30(2006).

Dwivedi, S., Vardhan, M., Tripathi, S., and Shukla, A. K., "Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection," Evolutionary Intelligence, vol. 13, no. 1, pp. 103.117, 2020.

Eiben, A. E., and Smith, J. E. 2003. Introduction to Evolutionary Computing. Springer.

Elmasry W., Akbulut A., and A. Zaim H., ‘Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic’, Computer Networks, 168, 107042, 2020.

Elmrabit N., Zhou F., Li F., and Zhou H., "Evaluation of Machine Learning Algorithms for Anomaly Detection," presented at the 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), Dublin, Ireland, Ireland 2020.

Enigo V. S. F., N. N. V. R., K. T. G., and D. S., "Hybrid Intrusion Detection System for Detecting New Attacks Using Machine Learning," presented at the 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020.

Farooqi, A. H., Khan, F. A., Wang, J. and Lee, S., A Novel Intrusion Detection Framework for Wireless Sensor Networks, Personal and Ubiquitous Computing, vol. 17, no. 5, pp. 907–919.

Farnaaz, N. and Jabbar, M., "Random forest modeling for network intrusion detection system," Procedia Computer Science, vol. 89, no. 1, pp. 213.217, 2016.

Fletcher, T. "Support Vector Machines Explained", UCL, 2009.

Gao, X., Shan, C., Hu, C., Niu, Z., & Liu, Z. (2019). "An adaptive ensemble machine learning model for intrusion detection." IEEE Access, 7, 82512-82521.

Hasan, H. and Tahir, N. M., 'Feature selection of breast cancer based on principal component analysis',6th International Colloquium on Signal Processing & its Applications, 1–4, 2010.

Hasan, M., Islam, M., Zarif, I.I., Hashem, M.A. "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches", Internet of Things, No.7,2019.

Haykin, S. "Neural networks: a comprehensive foundation", New York, USA: Macmillan Publishing, 1994.

He, K., Zhang, X., Ren, S., and Sun, J., 'Deep residual learning for image recognition', in Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778, 2016.

Hready, R.; Luger, G.; Maccabe, A.; Servilla, M., "The Architecture of Network Level Intrusion Detection System", Technical report, 1990, University of New Mexico, Department of Computer Science.

Hsu, Y.-F., He, Z., Tarutani, Y., & Matsuoka, M. (2019). "Toward an online network intrusion detection system based on ensemble learning." In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), IEEE, pp. 174-178.

Hu J., Liu C., and Cui Y., 'An improved CNN approach for network intrusion detection system', Int. J. Netw. Secur, 23, 4, 569–575, 2021.

Huddleston, S. H. and Brown, G. G., "Machine learning," in Informs Analytics Body of Knowledge, 2018.

Ibor, A. E.; Oladeji, F. A.; Okunoye, O. B., "A survey of cyber security approaches for attack detection prediction and prevention", International Journal of Security and Its Applications, vol. 12, no. 4, pp. 15-28, 2018.

Jasim, A. D. and Others, 'A survey of intrusion detection using deep learning in internet of things', Iraqi Journal For Computer Science and Mathematics,3,1,83–93, 2022.

Kaja, N., Shaout, A., and Ma, D., "An intelligent intrusion detection system," Applied Intelligence, vol. 49, no. 9, pp. 3235-3247, 2019.

Kendall, K., "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems", 1999.

Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J., "Survey of intrusion detection systems: techniques, datasets and challenges", Cybersecurity, vol. 2, no. 1, pp. 1-22, 2019.

Kita, E., (Ed.), Evolutionary Algorithm, InTech, pp 363.380,2011.

Kumar, V., Choudhary, V., Sahrawat, V., & Kumar, V. (2020). "Detecting Intrusions and Attacks in the Network Traffic using Anomaly based Techniques." In 2020 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, pp. 554-560.

Lin, J.; Chen, J.; Chen, C.; Chien, Y., "A Game Theoretic Approach to Decision and Analysis in Strategies of Attack and Defense", Secure Software Integration and Reliability Improvement, 2009. SSIRI 2009. Third IEEE International Conference on, On page(s): 75 – 81

Lu, T., Huang, Y., Zhao, W., & Zhang, J. (2019). "The Metering Automation System based Intrusion Detection Using Random Forest Classifier with SMOTE+ ENN." In 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), IEEE, pp. 370-374.

Maniriho, P. and Ahmad, T., "Analyzing the performance of machine learning algorithms in anomaly network intrusion detection systems," in 2018 4th International Conference on Science and Technology (ICST), 2018: IEEE, pp. 1-6

Meyer, M. L. B. and Labit, Y., "Combining Machine Learning and Behavior Analysis Techniques for Network Security," in 2020 International Conference on Information Networking (ICOIN), 2020: IEEE, pp. 580-583.

Miah, M. O., Khan, S. S., Shatabda, S., and Farid, D. M., "Improving Detection Accuracy for Imbalanced Network Intrusion Classification using Cluster-based Under-sampling with Random Forests," in 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019: IEEE, pp. 1-5.

Mirjalili, S., Lewis, A. "The whale optimization algorithm", Advances in engineering software, 2016, May 1; 95:51-67.

Mousavi, S. M., Majidnezhad, V., and Naghipour, A., "A new intelligent intrusion detector based on ensemble of decision trees," Journal of Ambient Intelligence and Humanized Computing, pp. 1-13, 2019.

Nadiammai, G. and Hemalatha, M., Effective Approach Toward Intrusion Detection System Using Data Mining Techniques, Egyptian Informatics Journal, vol. 15, no. 1, pp. 37–50, [2014]. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1110866513000418.

Onat, I. and Miri, A., "An Intrusion Detection System for Wireless Sensor Networks", WiMob â05, pp. 253.259, 2005.

Park, K., Song, Y., and Cheong, Y.-G., "Classification of attack types for intrusion detection systems using a machine learning algorithm," in 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), 2018: IEEE, pp. 282-286.

Patel, B., Somani, Z., Ajila, S. A., and Lung, C.-H., "Hybrid Relabeled Model for Network Intrusion Detection," in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018: IEEE, pp. 872-877.

Pombeiro, H. and Silva, C., "Intelligent Systems in Science and Information 2014," Stud. Comput. Intell., 2015.

Purcell, Stacy P.; Ross, Alan D.; Baca, Jim S.; Aissi, Selim; Kohlenberg, Tobias M.; Morgan, Dennis M., "Method and device for managing security events", December 2009

Qin, A.K., Huang, V.L., Suganthan, P.N. "Differential evolution algorithm with strategy adaptation for global numerical optimization", IEEE transactions on Evolutionary Computation, 2008, Sep 26;13(2):398-417.

Rafique, M. Ali, F., M., Qureshi, A. S., Khan, A., and Mirza, A. M., 'Malware classification using deep learning based feature extraction and wrapper based feature selection technique', arXiv preprint arXiv:1910. 10958, 2019.

Rajagopal, S., Kundapur, P. P., and Hareesha, K. S., "A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets," Hindawi, 2020, doi: 10.1155/2020/4586875.

Rani, K., Roopa, H., & Vani, V. (2019). "Prediction of Network Intrusion using an Efficient Feature Selection Method." In 2019 International Conference on Intelligent Computing and Control Systems (ICCS), IEEE, pp. 597-601.

Sah, G. and Banerjee, S., "Feature Reduction and Classifications Techniques for Intrusion Detection System," in 2020 International Conference on Communication and Signal Processing (ICCSP), 2020: IEEE, pp. 1543.1547.

Sai, Kirana, K.V.V.N.L., Kamakshi, D, R.N., Pavan Kalyana, N., Mukundinia, K., Karthi, R. "Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques", presented at the Third International Conference on Computing and Network Communications (CoCoNet'19), Procedia Computer Science 171, pp:2372–2379, 2020.

Sallam, A. A., Kabir, M. N., Alginahi, Y. M., Jamal, A., & Esmeel, T. K. (2020). "IDS for Improving DDoS Attack Recognition Based on Attack Profiles and Network Traffic Features." In 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), IEEE, pp. 255-260.

Shetty, Savita & Siddiqa, Ayesha, "Deep Learning Algorithms and Applications in Computer Vision", International Journal of Computer Sciences and Engineering, vol. 7, pp. 195-201, 2019.

Singh, S. and Banerjee, S., "Machine Learning Mechanisms for Network Anomaly Detection System: A Review," in 2020 International Conference on Communication and Signal Processing (ICCSP), 2020: IEEE, pp. 0976-0980.

Singh, P. and Venkatesan, M., "Hybrid approach for intrusion detection system," in 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), 2018: IEEE, pp. 1-5.

Smaha, S. E., "Haystack: An intrusion detection system", Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, pp. 37-44

Stiawan, D., Idris, M. Y. B., Bamhdi, A. M., & Budiarto, R. (2020). "CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection." IEEE Access, 8, 132911-132921.

Thaseen, S., Poorva, B., and Ushasree, P. S., "Network Intrusion Detection using Machine Learning Techniques," presented at the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020.

Uysal, E. İ., Demircioğlu, G., Kale, G., Bostanci, E., Güzel, M. S., and Mohammed, S. N., 'Network Anomaly Detection System using Genetic Algorithm, Feature Selection and Classification', 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 1–5, 2019.

Verma, S., Pant, M., Snasel V., 'A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems' IEEE Soft Computing Lab, DOI: 10.1109/ACCESS.2021.3070634, April 2, 2021.

Wang, Z. Li, W., Yan, Y., and Li, Z., 'PS--ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems', Expert Systems with Applications, 42, 22, 8881–8895, 2015.

Waskle S., Parashar L., and Singh, U., "Intrusion Detection System Using PCA with Random Forest Approach," in 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020: IEEE, pp. 803.808.

Yang, P., Zhou, J., Liu, L., Li, Y., 'A novel strategy of pareto-optimal solution searching in multi-objective particle swarm optimization (MOPSO) ' IEEE Computers & Mathematics with Applications, Volume 57, Issues 11–12, June 2009, Pages 1995-2000

Yihunie, F., Abdelfattah, E., and Regmi, A., "Applying Machine Learning to Anomaly-Based Intrusion Detection Systems," in 2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT), 2019: IEEE, pp. 1-5.

Zhang, J., Zhang,Y., "A spark-based ddos attack detection model in cloud services", International Conference on Information Security Practice and Experience, Springer, pp. 48–64, 2016

Zhou, Y., Cheng, G., Jiang, S., and Dai, M., "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Computer Networks, p. 107247, 2020.