
T.C.

ALTINBAŞ UNIVERSITY

Graduate School of Science and Engineering

Information Technologies

# A NEW COMPUTER NETWORK SYSTEM FOR INTRUSION DETECTION SYSTEM (IDS) DETECTION IN AD HOC NETWORK BASED ON OPTIMIZED DATA MINING TECHNIQUES

DUNYA IBRAHIM MOHAMMED MOHAMMED

Master's Thesis

Supervisor

Prof. Dr. Osman Nuri UÇAN

İstanbul, 2020

# A NEW COMPUTER NETWORK SYSTEM FOR INTRUSION DETECTION SYSTEM (IDS) DETECTION IN AD HOC NETWORK BASED ON OPTIMIZED DATA MINING TECHNIQUES

by

**Dunya Ibrahim Mohammed MOHAMMED**

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2020

The thesis titled "A NEW COMPUTER NETWORK SYSTEM FOR INTRUSION DETECTION SYSTEM (IDS) DETECTION IN AD HOC NETWORK BASED ON OPTIMIZED DATA MINING TECHNIQUES" prepared and presented by "Dunya Ibrahim Mohammed Mohammed" was accepted as a Master of Science Thesis in Information Technologies.

Prof. Dr. Osman Nuri UÇAN

Supervisor

Thesis Defense Jury Members:

| | | |
|---|---|---|
| Prof. Dr. Osman Nuri UÇAN | School of Engineering and Natural Sciences, Altinbas University | _____ |
| Asst. Prof. Dr. Çağatay AYDIN | School of Engineering and Natural Sciences, Altinbas University | _____ |
| Assoc. Prof. Dr. Adil Deniz DURU | Department of Sports and Health Sciences, Marmara University | _____ |

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

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

DUNYA IBRAHIM MOHAMMED MOHAMMED

# DEDICATION

First, I would like to thank Allah Almighty for the power of the mind, health, strength, guidance, knowledge, and skills to complete this study.

This thesis is wholeheartedly dedicated to my parents. There are no words to describe what you mean to me; there is nothing that I can repay for what you have done to me. I will continue to do my best to achieve your expectations.

Lastly, I dedicated this to the family, relatives, and friends who have been encouraging me during this study.

## ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Dr. Osman for all support during my study. Its great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

# ABSTRACT

## SECURITY ANALYSIS OF COMPUTER NETWORK USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES

MOHAMMED, DUNYA IBRAHIM MOHAMMED,

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Prof. Dr Osman Nuri UÇAN

Date: December / 2020

Pages: 57

In this study, Intrusion Detection System (IDS) Detection System in Ad Hoc Network based on Optimized Data Mining Techniques Based on Bayesian Optimization Algorithm. The data SVM is power data mining classifier used in various fields. The Bayesian Optimization Algorithm applied to optimize the parameters of SVM to presented high accuracy. Several evaluation techniques used to prove our method results. The obtained results compared with well-known researches presented in Intrusion Detection System (IDS) Detection field.

**Keywords:** Deep learning, IDS, CNN, KNN, SVM, PSO.

# ÖZET

## MAKİNE ÖĞRENME VE DERİN ÖĞRENME TEKNİKLERİ KULLANAN BİLGİSAYAR AĞININ GÜVENLİK ANALİZİ

MOHAMMED, DUNYA IBRAHIM MOHAMMED,

Yüksek Lisans, Bilişim Teknolojileri, Altınbaş Üniversitesi,

Danışman: Prof. Dr Osman Nuri UÇAN

Tarih: Aralık / 2020

Sayfa Sayısı: 57

Bu çalışmada Bayos Optimizasyon Algoritmasına Dayalı Optimize Edilmiş Veri Madenciliği Tekniklerine Dayalı Geçici Ağdaki İzinsiz Giriş Tespit Sistemi (IDS) Tespit Sistemi. SVM verileri, çeşitli alanlarda kullanılan güç veri madenciliği sınıflandırıcısıdır. Bayes Optimizasyon Algoritması, SVM parametrelerini yüksek doğrulukta optimize etmek için uygulandı. Yöntem sonuçlarımızı kanıtlamak için kullanılan çeşitli değerlendirme teknikleri. Elde edilen sonuçlar, İzinsiz Giriş Tespit Sistemi (IDS) Tespiti alanında sunulan iyi bilinen araştırmalarla karşılaştırılmıştır.

**Anahtar Kelimeler:** Derin Öğrenme, IDS, CNN, KNN, SVM, PSO.

# TABLE OF CONTENTS

**Page**

# LIST OF TABLES

**Page**

# LIST OF FIGURES

**Page**

# LIST OF ABBREVIATIONS

SVM : Support Vector Machine

NN : Neural Network

PSO : Particle Swarm Optimization Algorithm

CNN : Convolutional Neural Network

KNN : K-Nearest Neighbours Algorithm

# 1. INTRODUCTION

Theoretical background of computer viruses was introduced by Neumann in his seminal paper [1]," Theory of self-reproducing automata" in 1966. First practical viruses are written in between 1970 to 1980. First viruses were mostly harmless programs, written on a whim of programmers. Their developers were trying to investigate the conceptual limits of computers. All of them were easily cleaned [2].

First known computer virus is written by Bob Thomas. This virus copied itself between nodes of Arpanet– forerunner of Internet. When this program was written, the computer virus concept did not exist; nevertheless, it is generally accepted as the first computer virus [3]. Between 1980 and 1990, first wave of computer viruses started. In 1983, Cohen wrote the first article about computer viruses, giving both source code and theoretical background [4].

Cohen performed a number of experiments in different operating systems, and proved that virus concept is independent of programming language and operating system. Given suitable conditions, a virus can propagate very easily. Although first computer viruses and worms were written for mainframe and Unix systems, they found their fertile grounds in MS-DOS and windows systems. Most of these viruses were boot-sector viruses.

After 1990 to 2000, second wave of computer viruses started. Although viruses was dominant in first wave, with the advent of Internet and email, worms become more dominant. After 2000 to today, attacks of viruses and worms become a common phenomenon, and malware infestations become very common. All types of malware use vulnerabilities of applications and operating systems. Since writing programs and applications become more and more complex, vulnerabilities in programs will not decline.

There are different types of intrusion detection systems based on different approaches. There are two main differences between IDS and IDS signature-based behavior. There are several subcategories, depending on their implementation. Signature-based IDS credentials like Snort behave like antivirus software. I know of a list of attacks that can be used to learn new attack detection techniques. The system will generate an alert when a new action matches a known attack signature. The IDS behavior system works differently. Explore using different methods

(the most popular statistical analysis) that reflect normal behavior. Detects abnormal behavior, which generates an alert if the system has a normal behavior profile. Signature-based systems have become more popular these days because they are more reliable (fewer false positives), fewer false positives, and easier false positives. Behavioral VEDs are too specific (plus false negatives) to cause too many false positives and difficult to assess. At the same time, IDS behavior can identify new attacks, for example. Since the new attack is different from normal operation, this is a zero day attack. Signature-based IDS uses known attack signatures to identify attacks, so it cannot identify new attacks such as a zero-day vulnerability.

## 1.1 CONTRIBUTIONS OF THE THESIS

- Presented new IDS system combining the ant lion optimization algorithm with SVM, the ALO optimize the hyper parameters of the SVM to presented best classification rate.
- The proposed system presented accurate results when compared with several studies in literature.
- The system presented low execution time and increase the speed of SVM when dealing with large datasets.

## 1.2 RESEARCH QUESTIONS

In this thesis, several points were studied and investigated to deal with IDS attacks, these points are described below.

- How to detect the IDS in minimum cost and accurate performance?
- How to determine the classification technique to detect IDS attacks?
- How to increase the accuracy by combining SVM with optimization algorithms?
- How applied ALO to improve the SVM structure?

## 1.3 THESIS HYPOTHESES

- Data mining presented computer programs in a beneficial method to achieve intrusion detection.
- Effective systems with low cost developed using data mining techniques in different fields of computer security.
- Combine optimization algorithm with SVM and optimize the structure of the system by using ALO.

## 1.4 THESIS STRUCTURE

The letter includes five chapters prepared as follows:

- Chapter I: Introduction: IDs, problem statement, thesis goal and finally thesis structure.
- Chapter II: Literature Review: This chapter provides an overview of the work related to the detection of IDS anomalies.
- Chapter 3: Methodology: This chapter outlines the research methodology used in this thesis. Deep learning, CNN and the proposed method.
- Chapter 4: details of the implementation of the experiment and the results obtained for all proposed scenarios and comparison of results.
- Chapter 5: Conclusion and Future Work.

# 2. OVERVIEW

## 2.1 AD HOC NETWORK

Mobile Ad hoc network (MANET) is a set of wireless devices that communicate with each other together without central control. Recent research has focused on this point. twenty one MANET technology with inexpensive small wireless devices Communication devices. MANET can be used in various applications, such as Medicine, mobile courses, communication on the battlefield and disaster relief [5].

## 2.2 COMPUTER SECURITY

Cyber (Internet, workplace, intranet) creates a dangerous home for all governments and individuals to protect their hidden data or reputation. The goal is to receive people and equipment. It is important to mention that new research shows that bullying or unhappy employees, such as the Edward Snowden affair, pose an additional internal danger. Effortlessly, IT security must protect computer systems and data from damage, theft and illegal use. It is a procedure to prevent and detect illegal uses of your computer system [6]. Typically, people in the cloud protect a computer with additional related components, such as information security and cybersecurity. The only way to classify the similarities and changes between these fundamentals is to ask what is harmless.

For example: Information security protects data from illegal receipt, modification and deletion. The Security Security computer stores an autonomous machine which is safe, modern and refurbished. Cyber security differs from protecting computer systems by full interactive computer networks.

Avoid theft of data such as bank account numbers, credit card information, passwords, business documents or brochures, etc. Indispensable for today's communication, as many of our daily activities depend on secure data paths.

Computer data can also be used without authorization. An attacker could modify the program's source code or use photos or email accounts to create derogatory content such as pornographic images, fake accounts, and social attacks.

Malicious intentions can be a factor in computer security. Attackers often use your computers to attack other computers, websites, or networks and wreak havoc. Avenger hackers can destroy a person's computer system and cause data loss. DDOS attacks can be carried out to prevent access to websites in the event of a server failure.

The above factors mean that your data should be kept safe and confidential. Hence, your computer needs to be protected, and therefore computer security is vital.

## 2.3 P2P COMMUNICATION

P2P is the exchange of computing resources through direct exchange, as opposed to a conventional client-server network architecture, where one or more computers are designated as servers. This includes the ability to serve all customers on the network. Each P2P client receives service from the P2P network and serves the network at the same time. Shared resources in P2P networks can represent processor performance, memory, or files stored on computers. The two terms "P2P networks" and "P2P clients" should be defined to better understand P2P communication. A P2P network is a protocol with special rules that connects multiple P2P clients. A P2P client is simply a computer application that interacts with other clients on the network. Two types of packets are used in P2P exchanges: control packets and data packets [7]. The packets used to log on to the network perform searches, and the search results presented are control packets. Packages that contain portions of a common file transferred from one client to another are data packets.

### 2.3.1 P2P Networks

P2P communication in the current sense has become a popular application with the introduction of Napster [8] in 1999 by Sean Fanning. The Napster infrastructure is based on central index servers that manage a database with all the content on the network. Clients are currently connected at all times. If a user wants to find a file, they just search for Napster. The file request is sent to the central server through the Napster client. The server checks the "list of known files" and provides an Internet address to users who have a file for the requesting user. The Napster network peaked in February 2001: 29.4 million registered users and 2.79 billion shared files [9]. This rapid spread of Napster has led to complaints from Metallica, Dr. Dre and the American

Recording Industry Association. As a result, the servers at Napster were closed in late 2001. The use of central servers made indexing and search algorithms easier, but also allowed attorneys to easily detach the Napster network. The attorneys came to the server room and shut down the servers, which was the end of the Napster network.

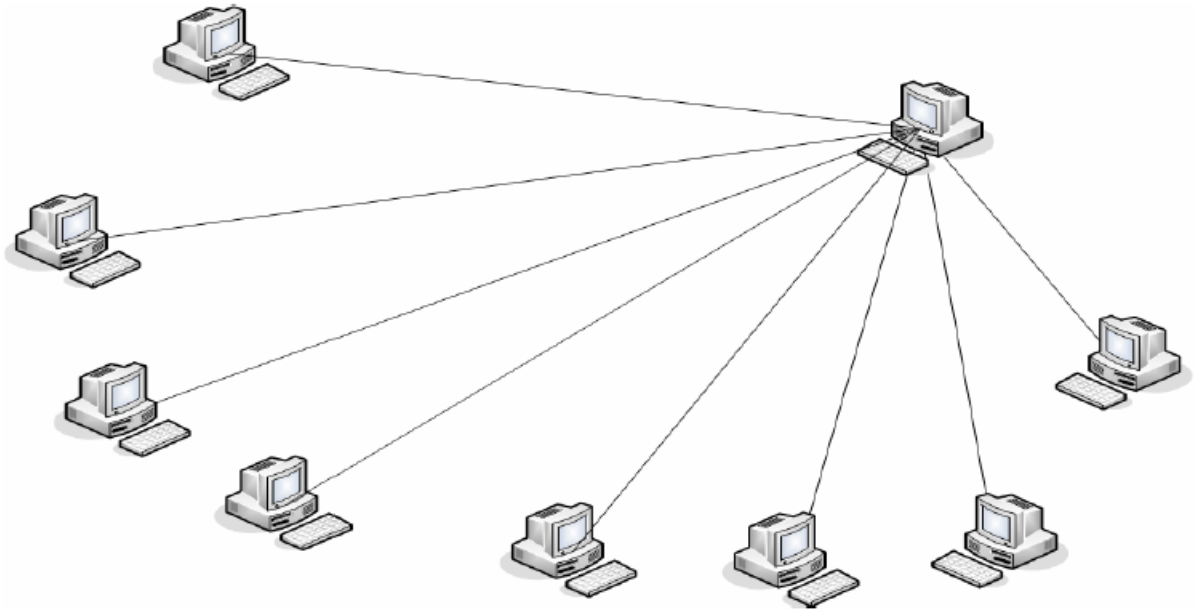

**Figure 2.1:** Centralized p2p networks topology [10].

The second generation of P2P protocols, Gnutella (with BearShare, Morpheus, Mutella, and Phex clients), was designed to correct a design error in Napster that led to a simple shutdown. In essence, Gnutella connects customers directly with other customers, resulting in a distributed / decentralized architecture. When a user starts the Gnutella client, he connects to a certain number of users who also connected to a certain number of clients. To search for a file, the Gnutella client requests a file from directly connected clients. When the client receives the file Query, search for a file in your database and respond to the query when the search managed. If the search in its file database was not successful, the client forwards request to nodes connected directly, with the exception of the applicant. This is a spread Search continues when a file value or TTL (Time to Live) is found Expiring Closing a Gnutella Network Is Not as Easy as Closing Napster because it requires disconnecting all communication clients. Gnutella, however, suffers from a poor search algorithm that creates an excessive level of control traffic.

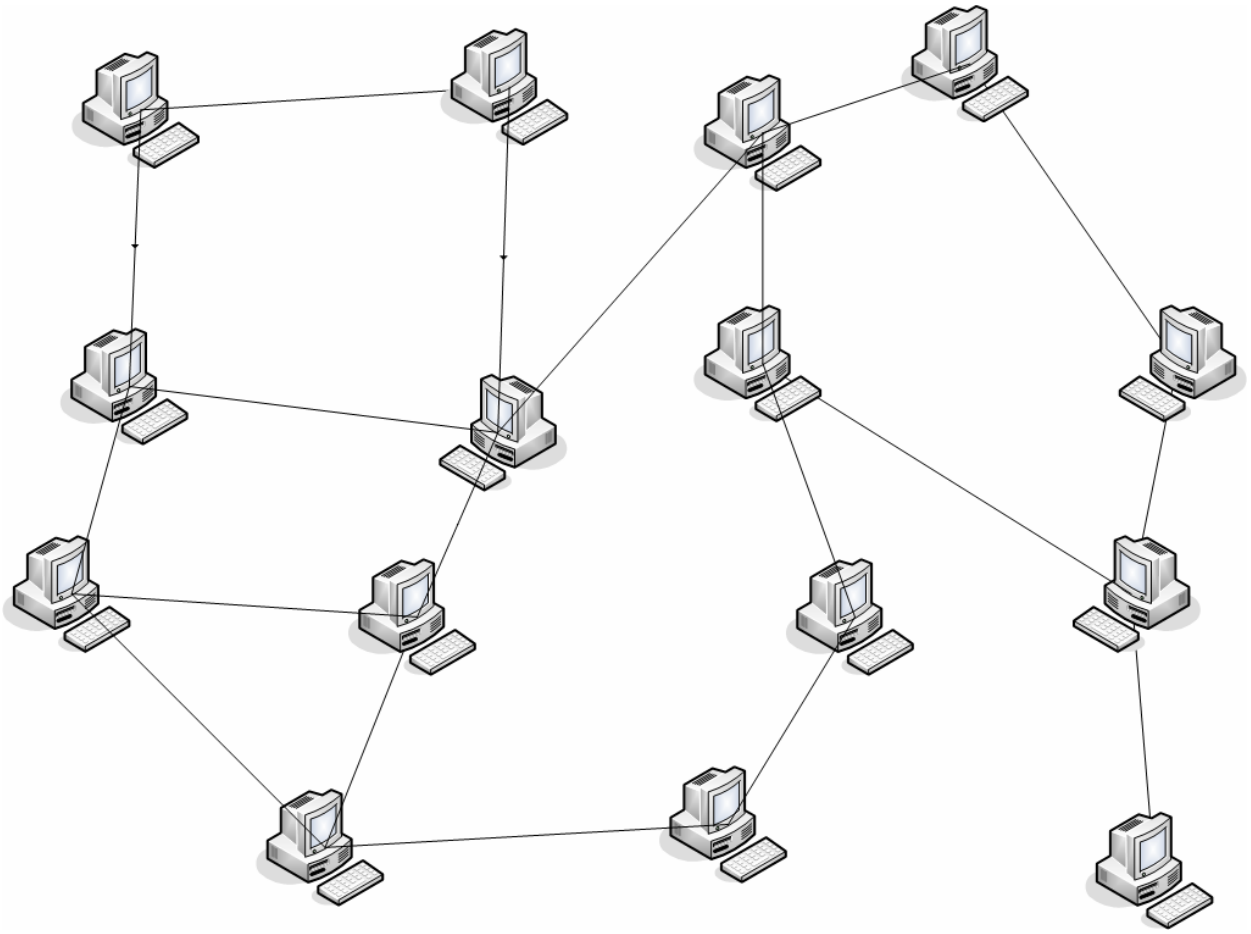

**Figure 2.2:** Decentralized p2p networks topology [10].

The inefficiency of distributed / decentralized architecture has caused more Improvements in the P2P network topology, resulting in the third generation of P2P Fast Track protocols. Soon The track combines the benefits of centralized and distributed / decentralized Topologies This hybrid network includes a series of supernodes that act similarly. to Napster index servers. These supernodes are selected by active users Fast Track client and high bandwidth Internet connections. These knots periodically index the databases of the users' files linked to them and share them information with other supernodes that reduce the amount of control traffic done online and increases the efficiency of search queries. Similarly The Direct Connect protocol [11] adds a chat function to connected users the same super node that makes the protocol more community oriented. The next step in P2P development is BitTorrent [12], which was developed primarily for Discourage leeches that only download files from P2P network without providing them none. The BitTorrent network uses a principle called tit-for-tat, which means giving Files

to get files. This is guaranteed by organizing the download speed of Client based on the number of shared files. Also the search procedure BitTorrent is very different from its ancestors. Use websites (site ads) to Distribute ".torrent" files. These ".torrent" files contain BitTorrent information Users who share all or part of a particular file. A user performs a web search to find ".Torrent" file of a file that interests you. After downloading the ".torrent" file I It has an approximate size of 50 KB, the user opens it through a BitTorrent client program. The BitTorrent client uses the ".torrent" file to contact a computer called a tracker, eg. hosts information about other clients where the entire file is affected (Seeds) or part of it. The tracker forms a swarm containing the network. All clients that download or upload the same file. Tracker regulates the trade of Files in the swarm. Web search function and an eye for an eye principle the use of the BitTorrent network grew rapidly.

### 2.3.2 P2P Traffic Detection

P2P traffic detection efforts have followed the development of P2P Claims The legal actions have been terminated by Napster. Furthermore, P2P taxpayers can be identified simply by checking the connection Traffic flow from Napster servers. Although an ineffective search process has created a second generation P2P network suddenly disappear, the idea of using decentralized topologies was inherited the next generation. After the introduction of third generation hybrid topologies. P2P network, deepening the exchange of P2P communication packets Studies to identify P2P packets. These investigations led to transportation Change of port number depending on discovery methods. Flow control. Packets and some data packets are created at a specific, static level of transport. Compounds (Table I). P2P users are identified by identifying these ports [13], [14]. More Firewalls have rules that prohibit traffic on certain P2P ports. P2P In response, client programmers have added dynamic port functionality in new versions which made the port-based discovery unusable. In new versions, when the user starts P2P client tries to contact super hosts with default Door. If the connection fails, the client changes the port randomly. At the user can also manually configure the work port.

**Table 2.1**: P2P clients and transport layer ports [15]

| P2P Client | Transport Layer Port(s) |
|---|---|
| eDonkey200 | 4661-4665 |
| FastTrack | 1214 |
| BitTorrent | 6881-6889 |
| Gnutella | 41170 |
| DirectConnect | 411-412 |

Port-based P2P detection has lost much of its use due to the invention of a dynamic port Mapping properties on P2P clients. P2P client ports offer default suggestions instead of acting individually on P2P detection.
The next step in P2P discovery was detection of P2P client signatures. Programs This was accomplished through data mining and package control. P2P mapping for specific channels that characterize it P2P Protocol [16]. Each P2P client program inserts some Requirement or response teams that allow other customers to understand the concept Communication The payload of each package is verified to find it Special lines. Therefore, the examined network traffic is intercepted or cloned. at a separate location for further investigation of signature-based P2P detection Methods a detailed examination of the signatures of the P2P client programs is described in [17]. The signatures used in the data packages are identified and listed in your work. We also displays some of their identified signatures to show how they look.

### 2.3.3 Intrusion Detection

Intrusion detection can be defined as the process of detecting malicious behavior. targeting the network and its resources [18]. Intrusion Detection Systems (IDS) Software, hardware, and rule groups that regularly monitor the network Elements and generate an alarm in case of intrusion. Network element under Inspection may be an application that runs on a computer with an operating system. Computers, a group of computers, or all computers on a network. In accordance with IDS analyzes the type of object being scanned and analyzes the relevant

protocols, for example, Log access and error logs and compare them with previously defined rules. If there is a match with If a hacking rule occurs, an alarm is generated. Anxiety can occur in various forms, for example, as a line in logs, email to the system administrator or short message service (SMS) to the mobile phone of members of the Network Operations Center (NOC). Twentieth A wide range of elements offers many IDS applications. On the other hand, according to the rule, all intrusion detection systems can be divided into three groups Studying process.

## 2.4 DATA MINING

### 2.4.1 Supervised Learning

Monitoring requests are invoiced, of course, in response to cataloging that requires accurate product classification or regression entries should a contribution to ongoing performance be required. Method for mutual monitoring of students, including logistic regression, NB, SVM, ANN, RF. The purpose of reciprocal regression and classification is to achieve a relationship or construct the input elements which do not efficiently produce effective output labels. Although we found that the "correct" information came only from the educational role, we were overwhelmed by the accuracy of our system. However, this does not mean that the information on the label is country-specific and accurate. Look. High or incorrect data designations clearly reduce the effectiveness of the method. This indicates the application of monitoring, but the point is the complexity of the system and the different and different efforts. A letter that means they are all together. System complexity refers to the complexity of the learning objectives with respect to the points of a polynomial. The appropriate system difficulty is usually determined based on detailed training. Complex systems should be chosen mainly if they lack functionality or if the function is not used regularly in situations where there are multiple options. In fact, the entire system is too suitable to be applied to a limited number of job facts. Retraining implies learning a target that corresponds to the actual brightness of the learning function, but does not facilitate the other functions. It is also recommended to use the learning function to see the effectiveness of a learning or organization. . . A boring marker draws a line between two facts. The model uses functions of all levels. In fact, we choose an additional linear function of less complexity [19].

The use of differential bias also speaks of a simplification of the system. In all systems we do our best about the bias that a particular member represents in error and adjust it so that the amount of error can vary between different groups of learners. So a system with a large change in bias can only use 20% of the period, while a system with a large change with a small bias can fail anywhere between 5 and 50% of the period. The period according to the required job. Your training. I find that the questions naturally go in contradictory directions. Usually an increase in preload leads to a slight change and vice versa. When building your system, the problem is the range of features you choose. However, you are ready to make your own decisions and familiarize yourself with locally. In general, switching to a system in which the accuracy of the criterion is relatively foolproof increases the bias. In some cases, it can get worse. In addition, in the instructions for creating a simplified system, the dimensions of changes in the system are specified according to the degree of completion of the learning function. Small, modest feature sets usually have to be learned through a system of minor modifications, and large feature sets often require changes to the system to accurately study feature architecture [20].

### 2.4.2 Unsupervised Learning

Unsupervised machine learning algorithms infer patterns from a dataset deprived of orientation to known, or branded, outcomes. Dissimilar supervised machine learning, unsupervised machine learning techniques cannot be straight realistic to a regression or a classification problem since you have no impression what the standards for the yield data strength be, making it unbearable for you to train the method the way you generally would. Unsupervised learning can in its place be applied for learning the fundamental building of the data [21].

Unsupervised machine learning purports to discover before nameless designs in data, but greatest of the period these decorations are deprived estimates of what supervised machine learning can attain. Moreover, meanwhile you don’t know what the results must be, there is no way to control how precise they are, creation supervised machine learning additional applicable to real-world problems.

The superlative time to apply unsupervised machine learning is when you don’t have data on target outcomes, like decisive a target marketplace for completely new creation that your

commercial has not ever sold beforehand. Though, if you are annoying to become a healthier sympathetic of your current customer base, supervised learning is the best method [22].

# 3. MATERIAL AND METHODS

## 3.1 FEATURE EXTRACTION TECHNIQUES

Entity extraction is a downsizing process in which the original raw dataset is reduced to more manageable groups for processing. A characteristic of this large amount of data is the large number of variables that require extensive use of the computation. This section introduces various methods of extracting functions.

### 3.1.1 Independent Component Analysis (ICA)

Independent component analysis (ICA) is a statistical and computational technique that identifies the hidden factors underlying a set of random variables, measurements, or signals. The CIA defines a generative model for multidimensional observable data, which is usually in the form of a large sample database. The model assumes that the data variables are linear mixtures of some unknown hidden variables and that the mixing system is also unknown. It is assumed that hidden variables are not external and independent of each other and are called independent components of the observed data. ICA can find these standalone components, also known as a source or agent. External CLI for principal component analysis and factor analysis. However, ICA is a more robust method and can be used to find the primary factor or source when these traditional methods fail completely. The data that ICA analyzes can come from a wide variety of applications including digital images, document databases, economic indicators, psychological metrics, and more. In many cases the measurements are provided as a series of parallel signals or time series. The term blind source sharing is used to describe this problem. Typical examples are the simultaneous audio signals that are recorded by several microphones, the interfering radio signals that are received by EEG mobile phones that have been recorded by different sensors, or the mixing of parallel time series in industrial processes [23], [24].

### 3.1.2 Isomap

Isomap means isometric view. Isomap is a spectral theory-based method for reducing non-linear dimensions, which requires the acquisition of lower dimensional geodesic distances. Isomap begins building a network of neighbors. Then use the distance from the chart to the approximate

geodesic distance between every pair of points. You can find low-dimensional insertions in your dataset by decomposing the eigenvalues of the geodesic distance matrix. For nonlinear variants, the Euclidean distance metric is valid only if the neighborhood structure can be approximated as linear. The Euclidean distance can be very difficult if there are holes nearby. On the other hand, if you measure the distance between the two points behind the collector, the distance between the two points will be closer. Let's understand this using a very simple 2D example [25],[26].

### 3.1.3 Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) derived from multi-dimensional vector representations based on analyzes of large bodies. However, LSA uses a fixed pop-up window (e.g. at the paragraph level) to perform a full body conjoint analysis. The factor analysis procedure (decomposition into singular values) is then applied to the exchange matrix to obtain a narrow range of indicators (generally 300-500). This size reduction ensures that words that are used in the same context have the same vectors even if they are not used in the same context. For example, "house" and "house" usually appear with "roof", but they rarely appear together in the same language. However, they have a similar vector representation in the LSA. LSA can group individual word vectors together to provide a measure of the semantics of a larger unit of text. Therefore, the meaning of a paragraph is the sum of the vector words in that paragraph. You can use this criterion to compare large blocks of text, for example to compare the meaning of sentences, paragraphs or entire documents. Figure 3.3 shows the LSA process [27].

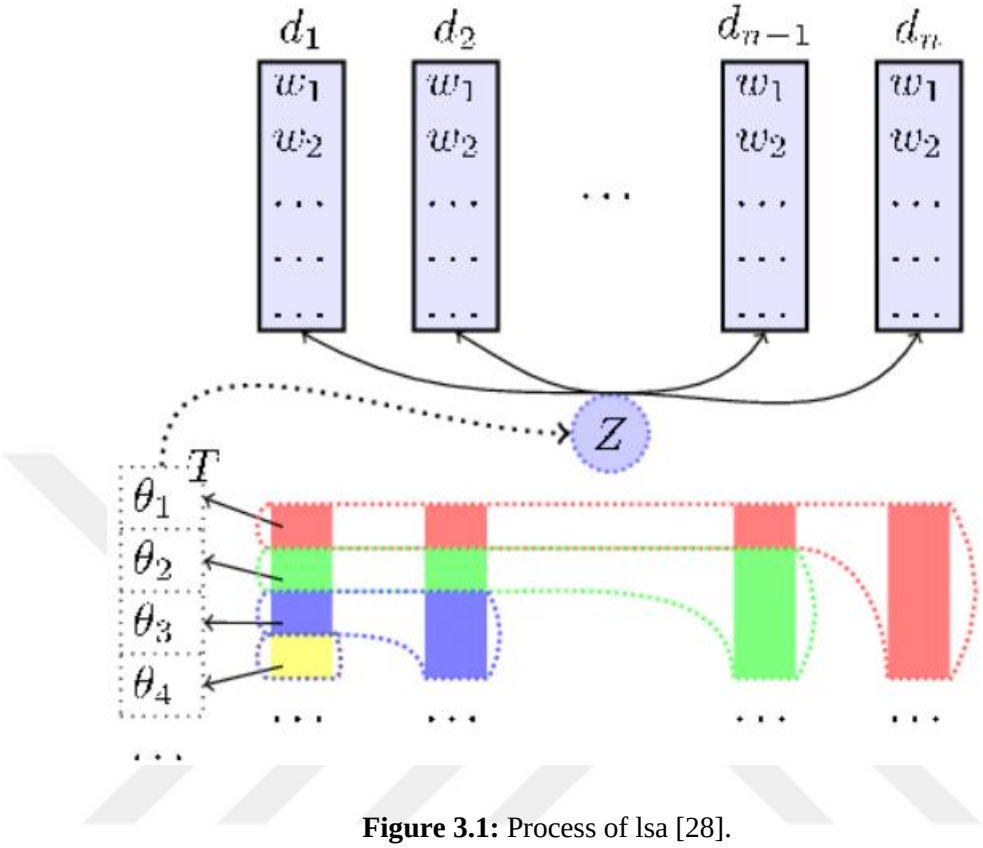


**Figure 3.1:** Process of lsa [28].

### 3.1.4 Partial least Squares (PLS)

PLS was originally developed using ultrasonic technology in the 1960s and is now used in many chemical measuring instruments. There are actually two versions of the PLS algorithm (PLS-1 and PLS-2). This can be confirmed under experimental conditions by simultaneous regression of the PLS and PCA type. The advantage of this method is that the experimenter can create a new explanatory variable that contains the same information as the original explanatory variable. Like polymerase chain reaction (PCR), PLS is one of the most widely used spectroscopic methods for multivariate synthesis. However, this is a stepwise process if there is no PLS regression step. Instead, PLS analyzes the spectral data and focuses on the data at the same time.

For example, to determine PC parameters, only spectra are taken into account in the original PCR analysis. When using PLS, analyte concentrations are also included in calibration samples. The factors are presented in such a way that it is possible to better clarify the evolution of the contained substances with the help of the PC, which must be determined [29].

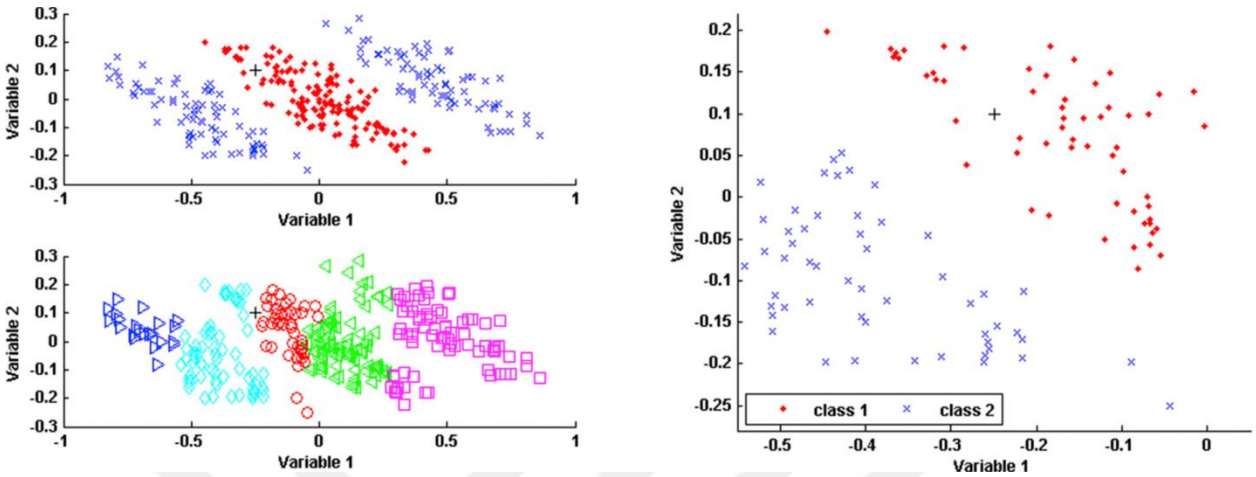


**Figure 3.2:** Pls [30].

### 3.1.5 Principal Component Analysis (PCA)

It is in Hotelling (1933) that was published in PCA Psychology of Education as we know it today. Placement shows intended use with factor analysis over the years. It has since been used in many other fields, including life sciences, natural sciences, and engineering. The main goal is to reduce the size of a multidimensional data set and clarify its interpretation by identifying fewer variables than summarizing a larger data set in some form [31]. The starting point for PCA is the matrix of correlation coefficients obtained from the original dataset. In fact, the justification of this method requires an estimation of the correlation of the measured variables on a continuous scale. In fact, for example, this variable can be completely continuous. They are separate, for example height, weight, or possibly as arithmetic variables. It is also handled well and can be called meters (i.e. major majors). This method can be used in cases where the correlation coefficient is not calculated using a scale variable. This analysis is not permitted, but it can give you a rough idea of the structure of your data.

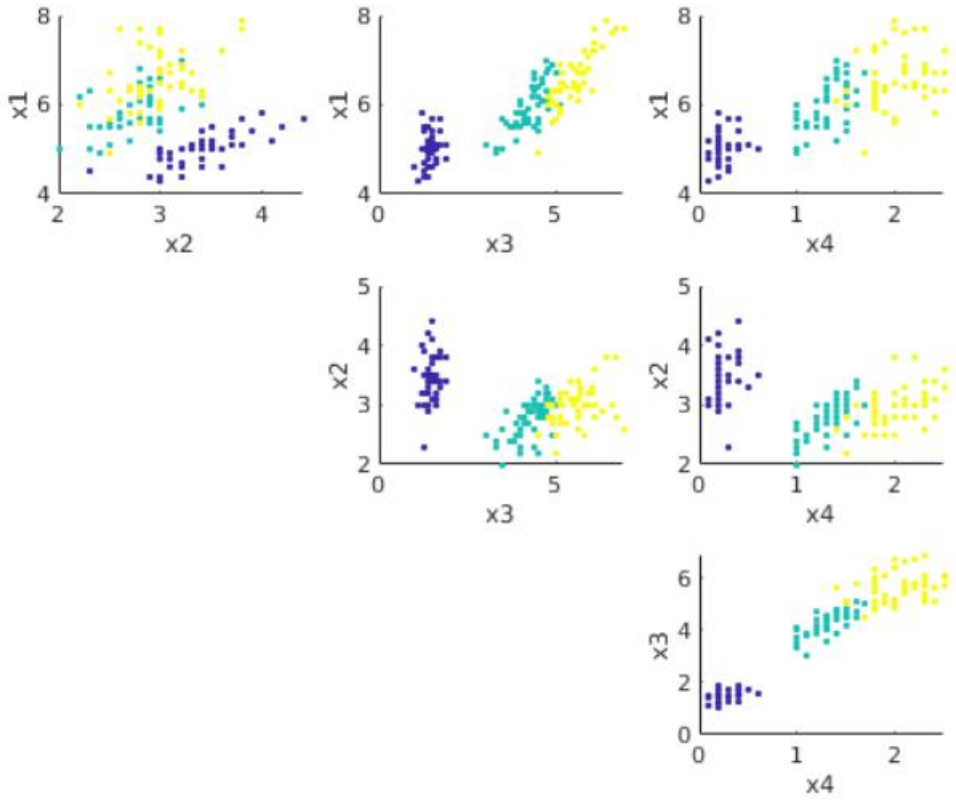


**Figure 3.2:** Pca process [32].

### 3.1.6 Auto-encoders

Auto-encoders are an uncontrolled learning technique in which we use neural networks to teach representation. In particular, we will design the architecture of a neural network to create a network bottleneck that will lead to a concise presentation of knowledge on the source data. If the input properties were independent of each other, this compression and subsequent retrieval would be a very difficult task. However, if there is some kind of structure in the data (i.e. correlations between the input properties), this structure can be studied and then used to force the entry through a bottleneck. Network restriction [33],[34].

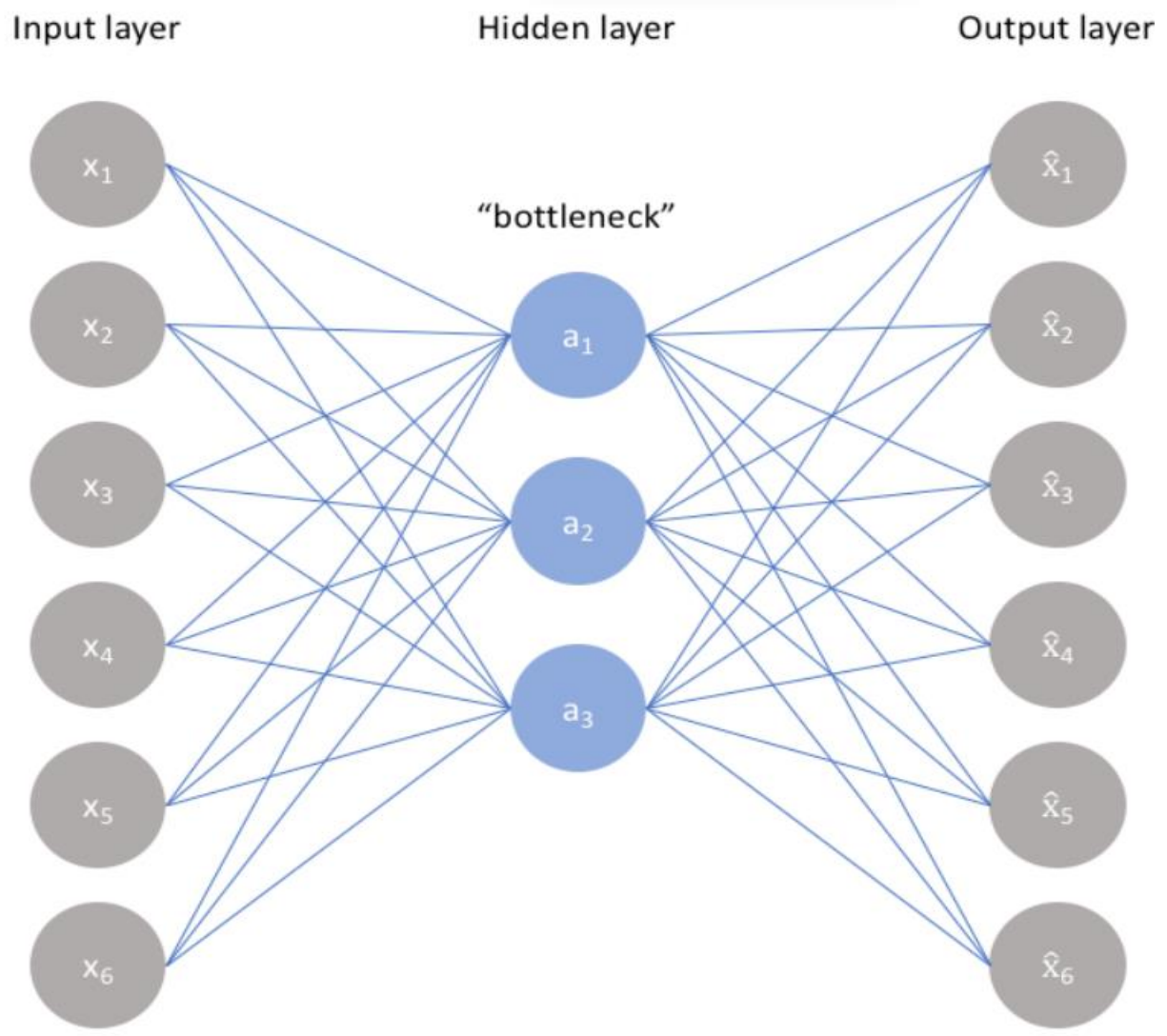


**Figure 3.3:**: Simple auto-encoders [35].

## 3.2 CLASSIFICATION TECHNIQUES

Classification is a central topic of machine learning that relates to machine learning to group data according to specific criteria. Classification is the process by which computers group data based on specified characteristics. This is called supervised learning. There is an automatic version of the classification called grouping, which has common functions on computers that allow you to group data when categories are not specified.

A typical classification example is spam detection. To write a spam filtering program, a programmer can train a machine learning algorithm using a series of emails that resemble spam

emails marked as spam. The idea is to create an algorithm that can learn the properties of spam from this tutorial so that you can filter spam when new messages arrive. In this section several classification techniques presented:

### 3.2.1 Deep Neural Network (DNN)

A Multi-layer Perceptron with growing quantity of layers is referred to as a Deep Neural Network with the time-honored subject being referred to as Deep Learning (DL) which has a focus on the lookup of excessive “depth” hierarchical illustration getting to know systems. This cross into deep fashions used to be pushed by using the trouble regular computing device learning (ML) techniques confronted which required deliberate statistics manipulations and preprocessing to attain applicable facets to be supplied as an enter to “shallow” fashions and gain good practical performance, that resulted in a requirement of analyzing statistics with a extraordinary deal of area information in order to extract appropriate enter vector options and with exponential increase in records dimension the effort turns into exponential brought to that the range of problems that are challenging to formalize efficaciously such as consciousness of spoken phrases or recognition of faces in images. Artificial neural networks (ANNs) Inspired by the processing of information in biological systems and distributed communication nodes. RNA is very different from a biological brain. Neural networks are usually static and symbolic, while the biological brain of most organisms is dynamic (plastic) and the like. The reason deep learning is poor quality is that it uses multiple layers in the network. The first task showed that a linear sensor cannot be a general classifier, and an invisible layer can have an infinite width of networks of admissible non-polypolic functions. Deep learning is a modern option that handles unlimited levels of limited size and allows for practical application and optimal implementation while maintaining theoretical diversity under normal conditions. In deep run, levels are also very different from biofeedback and communication-oriented models, which can be patchy and have parts “structured” for efficiency, forgiveness, and understanding.

.

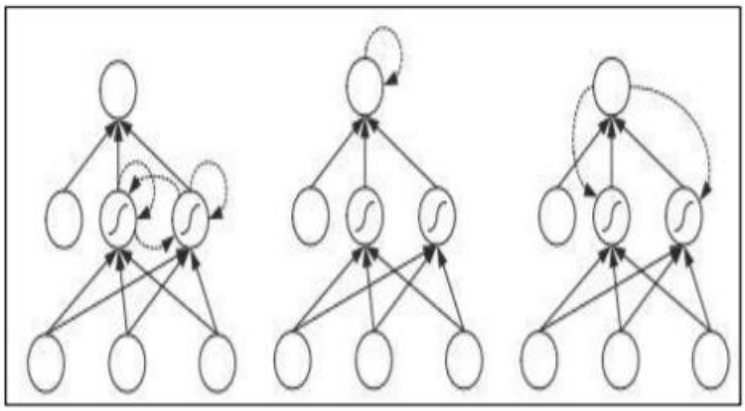

**Figure 3.4:** Recurrent neural network [37].

### 3.2.2 Support Vector Machine (SVM)

The goal of SVM is to find a hyperplane in an N-dimensional space (N is the number of faces) that uniquely classifies data points. You can select several imaginable hyperplanes to discrete the two lessons of information opinions. Our goal is to find the level with the maximum reach, or H. The maximum distance between the data points of the two classes. The edge distance optimization offers a certain gain and enables a more reliable classification of future data points [38]. In Figure 3.6 above, we notice that there are two classes of observations: the blue points and the purple points. There are tons of ways to separate these two classes as shown in the graph on the left. However, we want to find the "best" hyperplane that could maximize the margin between these two classes, which means that the distance between the hyperplane and the nearest data points on each side is the largest. Depending on which side of the hyperplane a new data point locates, we could assign a class to the new observation. It sounds simple in the example above. However, not all data are linearly separable. In fact, in the real world, almost all the data are randomly distributed, which makes it hard to separate different classes linearly.

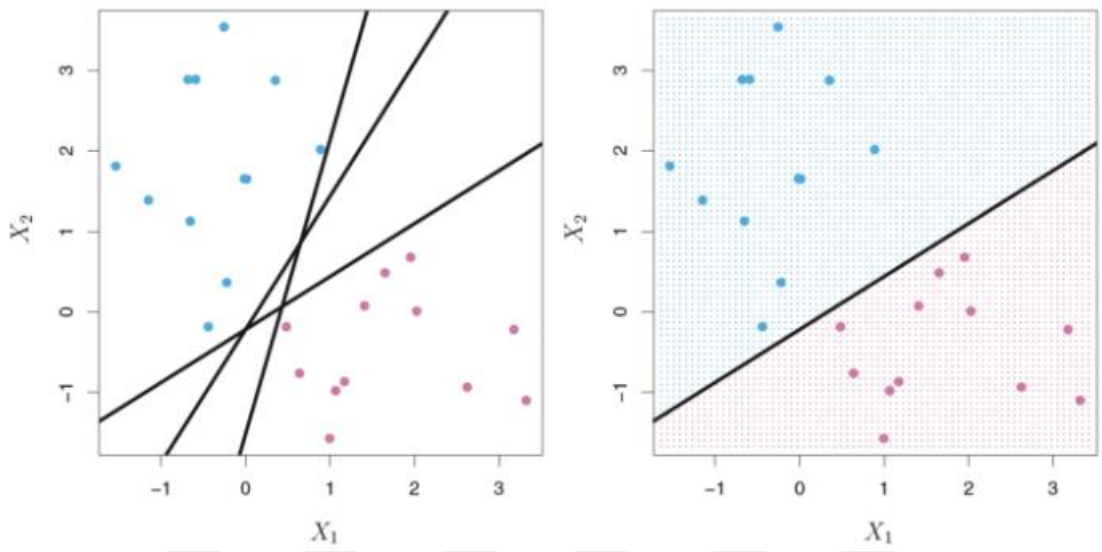


**Figure 3.5:** Hyperplanes in 2d and 3d feature space [39].

Hyperplanes are choice limits that can be used to classify information opinions. Data points on either lateral of the hyperplane can be assigned to dissimilar lessons. The size of the hyperplane also depends on the amount of functions. When the amount of input purposes is 2, the hyperplane has only one row. When the amount of input topographies is 3, the hyperplane becomes two-dimensional. It's hard to imagine if the number of functions exceeds 3.

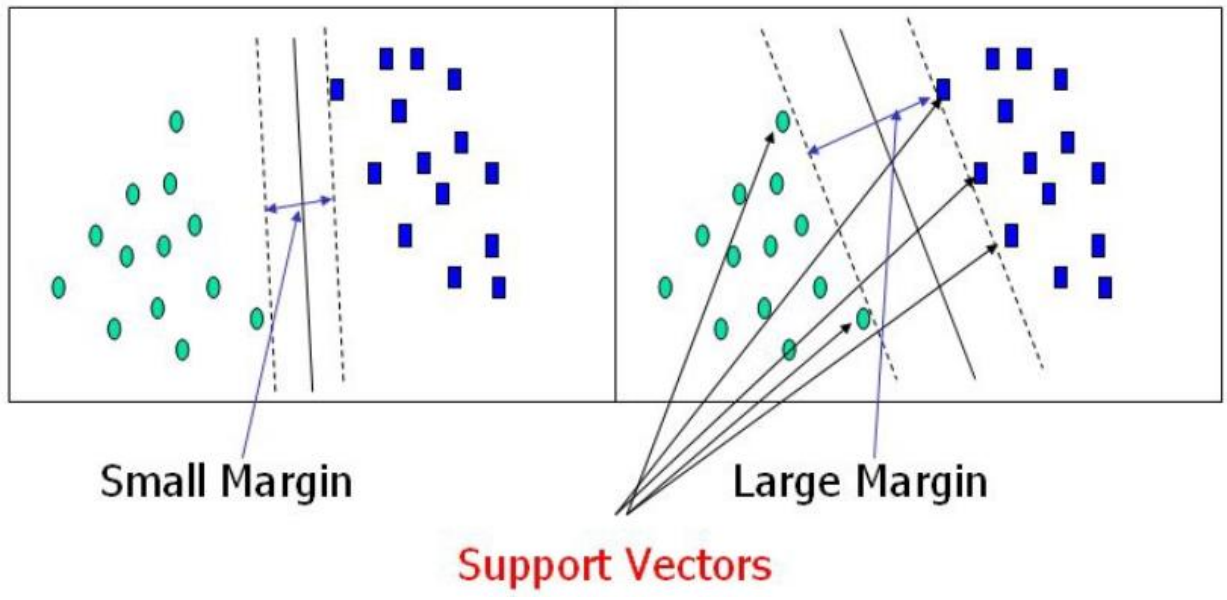


**Figure 3.6:** Margin [39].

### 3.2.3 Decision Trees (DT)

DT is a prediction model that expresses the characteristic space in a recursive partition into subspaces and forms the basis of prediction. DT is a direct root tree. In DT, the nodes with protruding edges are internal nodes. All other nodes are DT nodes or leaves. DTs are classified according to a hierarchical set of characteristic decisions. The decision made at the internal node is the distribution criterion. In DT, each leaf is assigned to a class or its probabilities. Small deviations in the drive unit lead to different units leading to different DTs. Therefore, the error contribution due to variance is important to DT. Collaborative learning, described in the next section, helps reduce errors due to deviations [40].

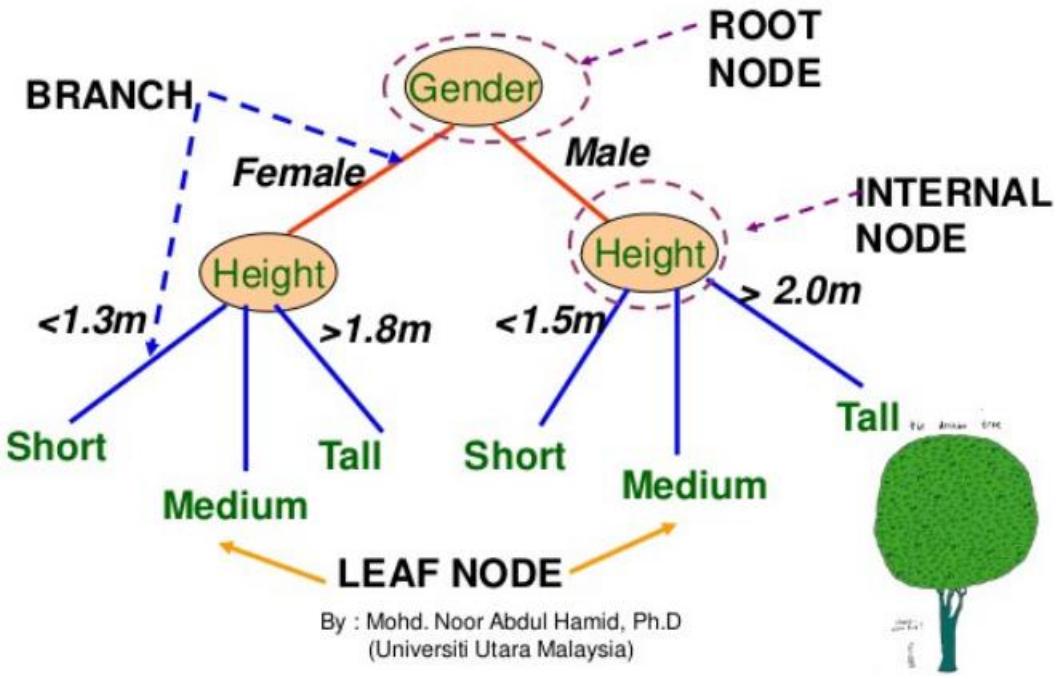


**Figure 3.7:** Decision tree [41].

### 3.2.4 Naïve Bayes Classifiers

This classification method is based on the statute and requires independence between 予 測 子. Simply put, it is assumed that the existence of a certain characteristic of the naive class of Bayesian classifiers is independent of the availability of other features. Apples, for example, are red berries that are about 3 inches in diameter. Regardless of whether these signs are interrelated

or dependent on other signs, these signs independently affect the likelihood that all of these fruits are apples and thus "slob". Known as the Naive Bayesian model, it's easy to create and is especially useful for large amounts of data. The simple Bayesian approach not only overcomes simplicity but also very complex classification methods [42].

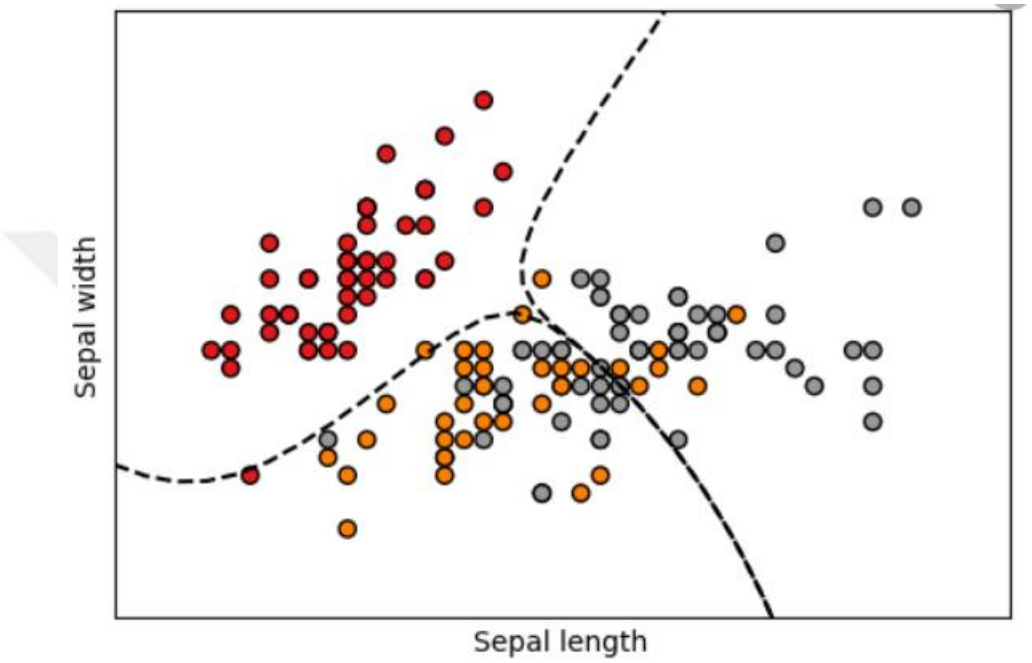


**Figure 3.8:** Naïve bayes classifiers [43].

### 3.2.5 K-Nearest Neighbors (KNN)

KNN is a simple and easy-to-manage machine learning algorithm that can solve classification and regression problems. The ANN algorithm assumes that these objects are very close to each other. I mean these things are related to each other. In most cases, similar data points are located next to each other, so take a look at the image above. The ANN algorithm assumes that this assumption is a very valid and useful algorithm. KNN calculates the distance between points on a graph and reflects the idea of similarity (also called distance, proximity, or proximity) through specific calculations that the child can learn [44].

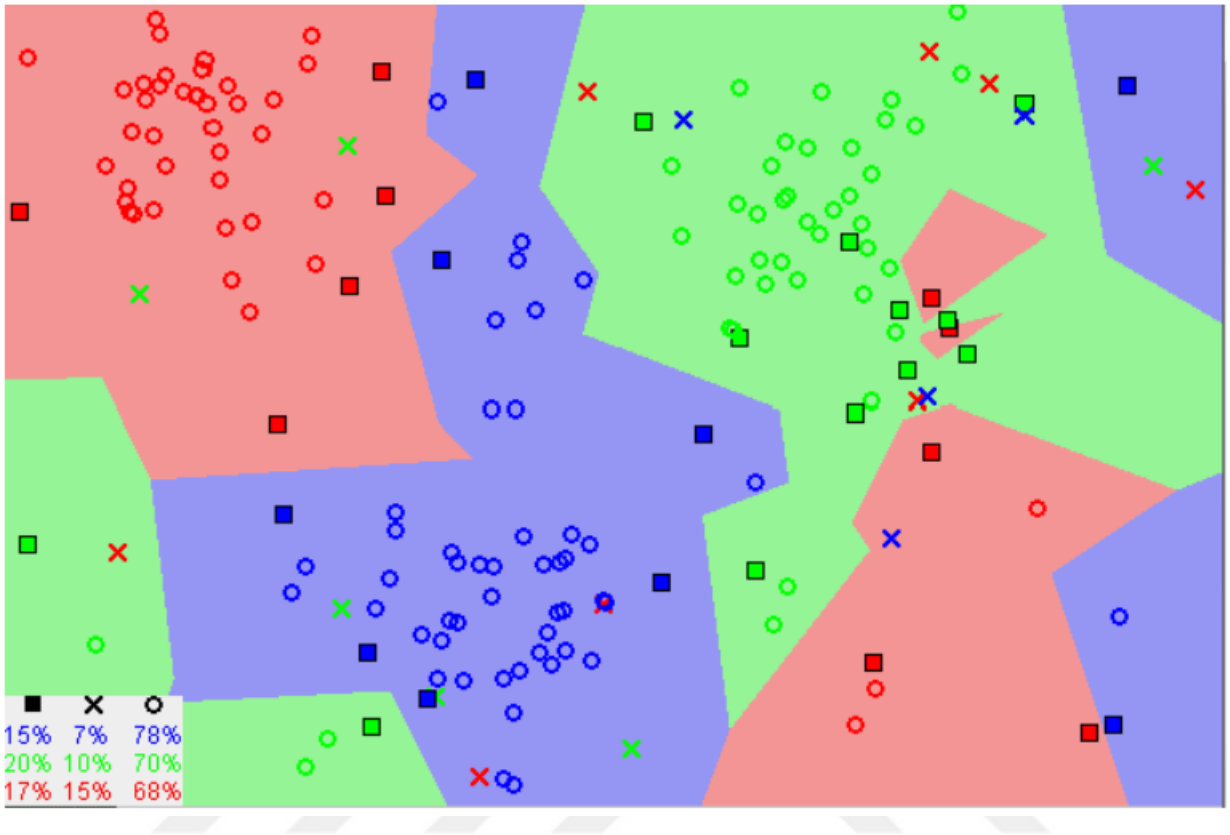


**Figure 3.9:** Knn operations [45].

### 3.2.6 Gradient Descent

In gradient descent, Batch is the total number of samples that you will use to calculate the gamut in one iteration. So far we've assumed this is the complete package. When working with Google, newspapers typically have billions, if not hundreds of billions of examples. In addition, Google records often contain multiple functions. So the party can be huge. It can take a long time to calculate the jackpot. A large data set with randomly selected examples can contain redundant data. Indeed, as the size of the packets increases, the repetition potential increases. Some redundancy can be useful to compensate for noise gradients, but large beams generally do not have much higher predictive value than large beams. What if we could get the correct mean gradient for a much smaller calculation? By choosing random examples from our dataset, we were able to (albeit aloud) estimate a high mean from a much smaller average. Stochastic Regression (SGD) takes this idea to the extreme: Use an example (batch size 1) to iterate. SGD

works with a large number of iterations but is very powerful. The term "stochastic" indicates that the example that contains each batch is selected at random [46], [47].

### 3.2.7 Random Forests

This assumes that you are familiar with creating your own classification tree. Many tree species grow in random forests. Place the input vector in each window in the forest to classify new features based on the input vector. Each tree offers a classification and the tree can "vote" on this category. The forest chooses the rank with the highest number of votes (among all trees in the forest). To understand and use different parameters, it is helpful to know more about how they are calculated. Most of the setup is based on 2 data objects generated from a random forest. If the training set is removed from the current tree using an alternate model, approximately one-third of the cases will be excluded from the sample. This collected data is used to objectively assess classification errors when adding trees to the forest. It is also used to get estimates of various values. After each tree is created, all tree data and calculated similarities are run for each pair of observations. If two states occupy the same endpoint, the convergence increases by one. At the end of the analysis, it is normalized by dividing by the number in the similarity tree. The technique is used to create a small and attractive data representation that replaces missing data and detects anomalies [48].

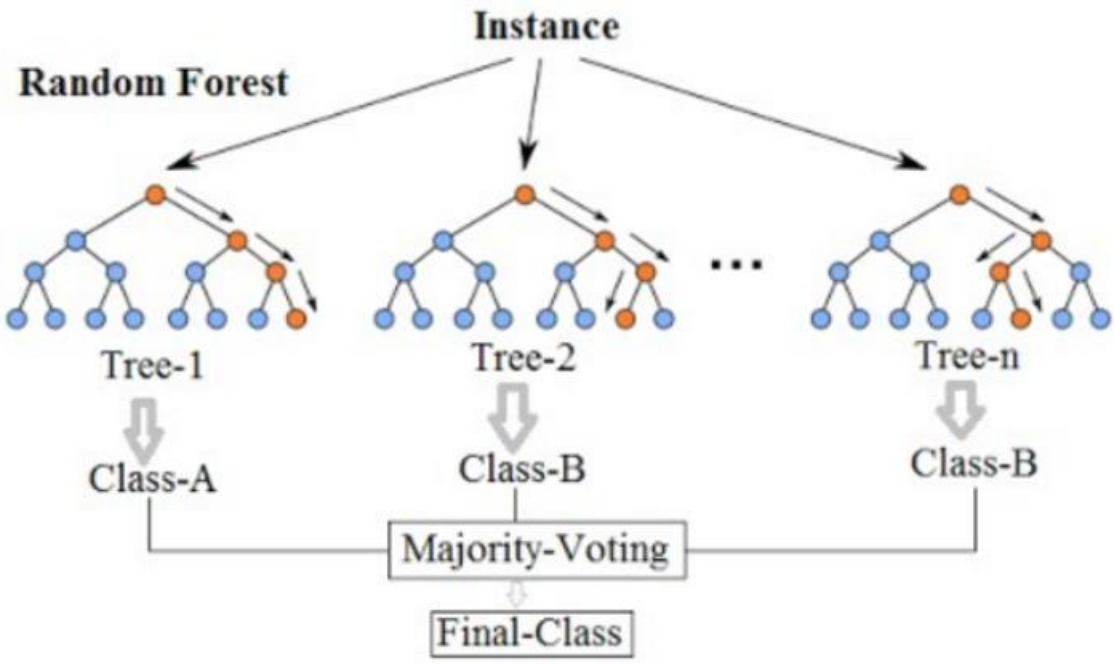


**Figure 3.10:** Random forests [49].

### 3.2.8 Discriminant Analysis (DA)

The discriminant analysis (DA) is mentioned often. Compare with other credit check methods. This approach is a classification. The criteria are based on instances with known categories and predictions of unknown categories based on these criteria. DA is parametric or non-parametric. Parametric AD builds a model with some assumptions about the distribution of instances (e.g. a normal distribution). However, the model is unbalanced due to the unobservable distribution. This parametric DA is not very common and DA performance is poor compared to nonlinear DA problems. In contrast, there is a non-parametric AD in the context of a credit rating. This approach examines the distribution of instances using nonparametric constructs and methods. Classification criteria Therefore the results of the nonparametric AD are more robust [50].

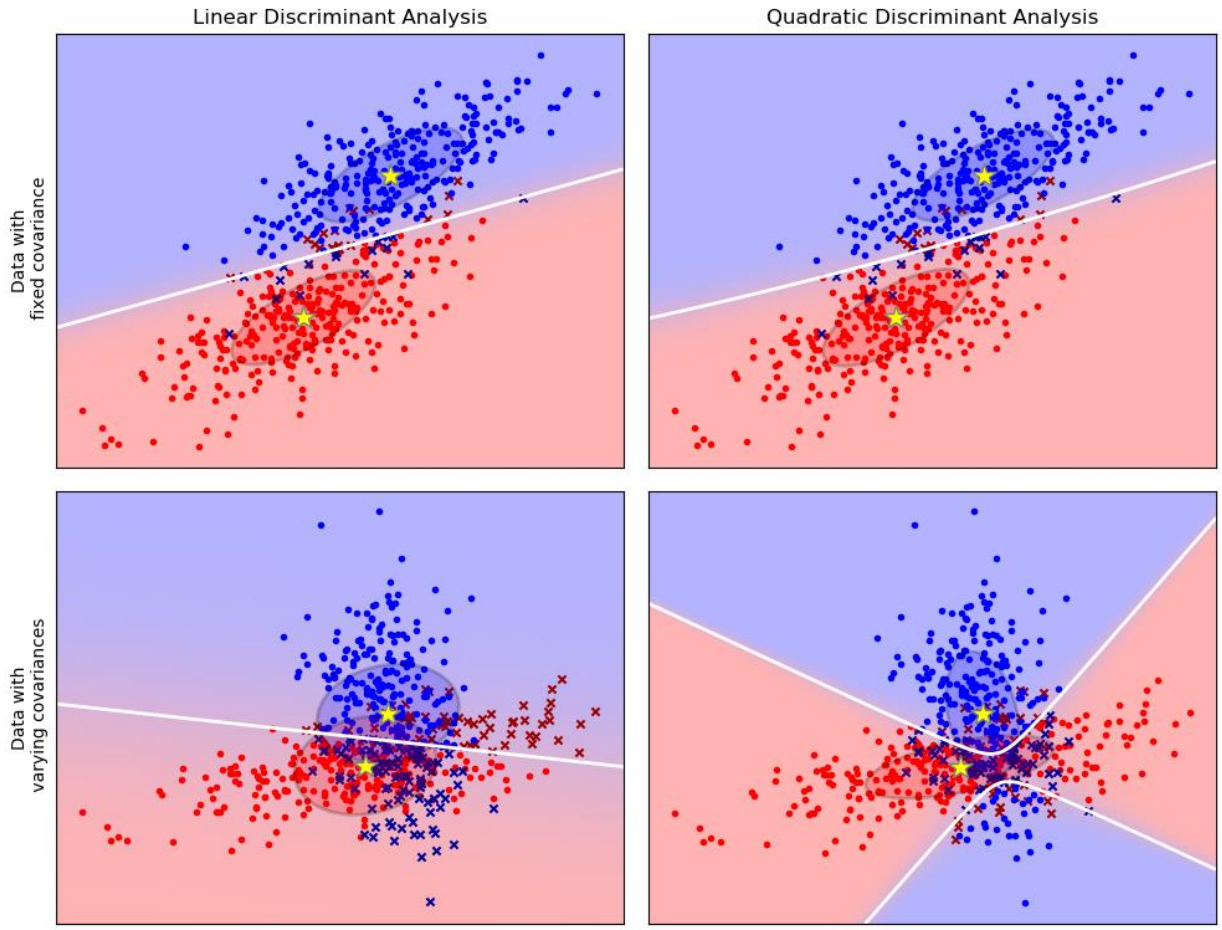


**Figure 3.11:** Discriminant analysis [51].

## 3.3 OPTIMIZATION ALGORITHMS

Optimization is everywhere and therefore it is a broadband paradigm. Applications We are still testing almost all technical and industrial applications. Optimize anything to minimize and maximize costs and energy consumption. Profit, production, productivity and efficiency. In fact, resources, time and money are always limited. Therefore, in practice, optimization is much more important. Optimal use of all kinds of available resources requires a paradigm shift in scientific thinking, since most real-world applications are much more complex factors and parameters that affect the behavior of the system. Modern technical design relies heavily on computer modeling. These This leads to further optimization difficulties. Growing demand for precision electronics. The growing complexity of projects and systems leads to a modeling process. More and more time. Evaluation of a project in many areas of mechanical engineering. This can take several

days or even weeks. Any method that can speed up the simulation. The optimization process and time can save time and cost [52], [53].

### 3.3.1 Bayesian Optimization Algorithm (BOA)

Bayesian optimization is a sequential design strategy for global optimization of black-box functions that does not assume any functional forms [54,55]. It is usually employed to optimize expensive-to-evaluate functions. Bayesian optimization is typically used on problems of the form $max_{x \in A} f(x)$, where $A$ is a set of points whose membership can easily be evaluated. Bayesian optimization is particularly advantageous for problems where $f(x)$ is difficult to evaluate, is a black box with some unknown structure, relies upon less than 20 dimensions, and where derivatives are not evaluated. Since the objective function is unknown, the Bayesian strategy consists of treating it as a random function and placing it a priori on it. The previous one captures ideas about the behavior of a function. After collecting estimates of the functions considered data, the previous ones are updated to create a background distribution for the objective function. After the submission, a data collection function is created (often called a padding criterion) that in turn determines the next polling point.

## 3.4 IDS DETECTION FRAMEWORK BASED SVM AND BOA

In this section, new framework based on SVM based on PCA and BOA. The BOA applied to optimize the structure of the SVM and for feature selecting.

The main challenge of this study is how can we present the problem as objective function which can be understand by optimization algorithm BOA. The objective function of this problem presented in equation 3.1.

$$Objective = \alpha * ErrorRate + (1 - \alpha) * \frac{\#SF}{\#All\_F} \quad (3.1)$$

The function 3.1 is used for selecting features which applied as feature selection objective function. Furthermore, ACC applied as to increase the Acc then, our method introduced as multi objective function.

$$Acc = \frac{TP+TN}{Total} \quad (3.2)$$

Where ACC represented the accuracy of the system, TP presented the true positive, TN represented the true negative, and Total represented the total estimation of the system TP, FN, FP, and TN.

This mean maximum is best which BOA remain trying the to find the best parameters right to find the maximum $Acc$.

The details of the proposed method presented in the figure 3.6.

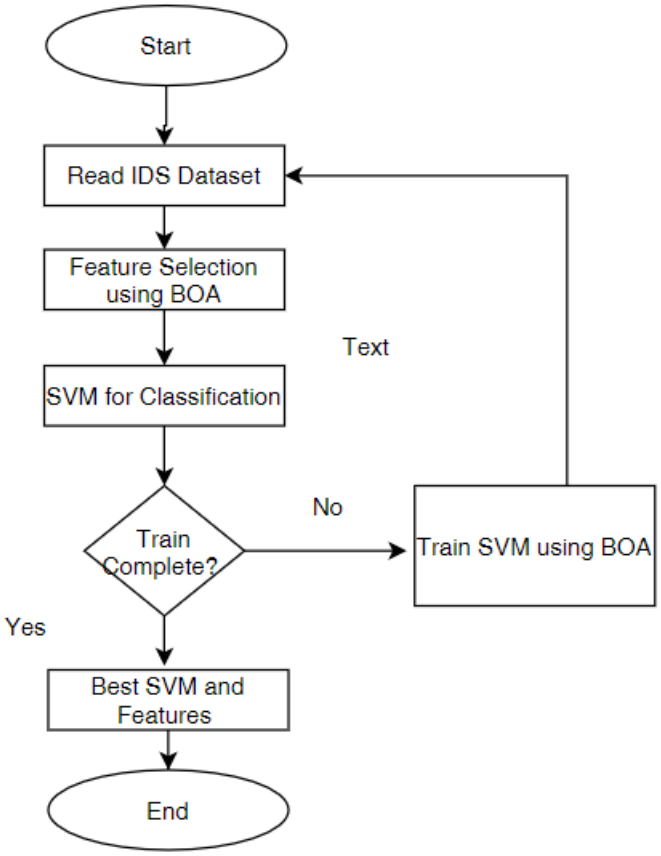


**Figure 3.12:** Flowchart of SVM Based BOA.

# 4. EXPERIMENTS AND DISSCUSION

Numerous tests are implemented in many scenarios, the tests and the outcomes were measured applying different capacities, the output of many tests was compared and the outcomes were highlighted.

The experimental executed by applying MATLAB2019 as programming. The database features that clarified in the chapter 3 applied as input to the future scheme that written by using MATLAB2019.

## 4.1 DATASET

The dataset [59] consists from 49 features as input and one column as target which represent the case there is IDS attack or not. The dataset consists from 257673 instances which divided into training and testing parts 175,341for training and 82,332 for testing. feature of dataset presented in Table 4.1. Furthermore the dataset consist from 9 attacks types: Analysis, Backdoors, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, and Worms. Moreover, the normal case also added to the classes and then become 10 classes.

**Table 4.1:** Dataset features [59]

| SNo. | Name | Type | Description |
|---|---|---|---|
| 36 | is_sm_ips_ports | Binary | "If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal thenthis variable takes value 1 else 0" |
| 39 | is_ftp_login | | If the ftp session is accessed by user and password then 1 else 0. |
| 49 | Label | | 0 for normal and 1 for attack records |
| 7 | dur | Float | Record total duration |
| 15 | Sload | | Source bits per second |
| 16 | Dload | | Destination bits per second |
| 27 | Sjit | | Source jitter (mSec) |
| 28 | Djit | | Destination jitter (mSec) |
| 31 | Sintpkt | | Source interpacket arrival time (mSec) |
| 32 | Dintpkt | | Destination interpacket arrival time (mSec) |
| 33 | tcprtt | | "TCP connection setup round-trip time, the sum of synack and ackdat." |
| 34 | synack | | "TCP connection setup time the time between the SYN and the SYN_ACK packets." |
| 35 | ackdat | | "TCP connection setup time the time between the SYN_ACK and the ACK packets." |
| 2 | sport | Integer | Source port number |
| 4 | dsport | | Destination port number |
| 8 | sbytes | | Source to destination transaction bytes |
| 9 | dbytes | | Destination to source transaction bytes |
| 10 | sttl | | Source to destination time to live value |
| 11 | dttl | | Destination to source time to live value |
| 12 | sloss | | Source packets retransmitted or dropped |
| 13 | dloss | | Destination packets retransmitted or dropped |
| 17 | Spkts | | Source to destination packet count |
| 18 | Dpkts | | Destination to source packet count |
| 19 | swin | | Source TCP window advertisement value |
| 20 | dwin | | Destination TCP window advertisement value |
| 21 | stcpb | | Source TCP base sequence number |
| 22 | dtcpb | | Destination TCP base sequence number |
| 23 | smeansz | | Mean of the ?ow packet size transmitted by the src |
| 24 | dmeansz | | Mean of the ?ow packet size transmitted by the dst |
| 25 | trans_depth | | Represents the pipelined depth into the connection of http request/response transaction |
| 26 | res_bdy_len | | Actual uncompressed content size of the data transferred from the servers http service. |
| 37 | ct_state_ttl | | No. for each state (6) according to specific range of values for source/destination time to live (10) (11). |
| 38 | ct_flw_http_mthd | | No. of flows that has methods such as Get and Post in http service. |
| 40 | ct_ftp_cmd | | No of flows that has a command in ftp session. |
| 41 | ct_srv_src | | No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26). |
| 42 | ct_srv_dst | | No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26). |
| 43 | ct_dst_ltm | | No. of connections of the same destination address (3) in 100 connections according to the last time (26). |
| 44 | ct_src_ ltm | | No. of connections of the same source address (1) in 100 connections according to the last time (26). |
| 45 | ct_src_dport_ltm | | No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26). |
| 46 | ct_dst_sport_ltm | | No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26). |
| 47 | ct_dst_src_ltm | | No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26). |
| 1 | srcip | Nominal | Source IP address |
| 3 | dstip | | Destination IP address |
| 5 | proto | | Transaction protocol |
| 6 | state | | Indicates to the state and its dependent protocol |
| 14 | service | | "http ftp smtp ssh dns ftp-data |
| 48 | attack_cat | | "The name of each attack category. In this data set nine categories *eg* Fuzzers Analysis Backdoors DOS exploits Generic Reconnaissance Shellcode and Worms" |
| 29 | Stime | Timestamp | record start time |
| 30 | Ltime | | record last time |

## 4.2 EVALUATION

As shown in figure 4.1 and figure 4.2 the features of normal and abnormal cases are ploted and saved in different folders. The CNN applied to extracted best features from input features then the extracted features wired to the SVM classifier.

In this method, random sub-sampling validation is track for a secure amount of K iterations. Throughout apiece iteration it used random sampling minus spare, in instruction to choice a secure amount of S examples that brand up the examination set and are excepted from the training procedure of the perfect.

Several evaluation parameters are calculated to evaluate the performance of the proposed method. These parameters are shown in Eq(4.1)-(4.5)

$$\text{TPR} = \frac{TP}{(TP + FN)} \tag{4.1}$$

$$\text{SPC} = \frac{TN}{(FP + TN)} \tag{4.2}$$

$$\text{PPV} = \frac{TP}{(TP + FP)} \tag{4.3}$$

$$\text{NPV} = \frac{TN}{(TN + FN)} \tag{4.4}$$

$$\text{ACC} = \frac{(TP + TN)}{(P + N)} \tag{4.5}$$

These parameters calculated to validate the proposed method as shown in the Table 4.2.

**Table 4.2:** Statistical parameters

| Results | Results |
|---|---|
| Sensitivity | 0.9314 |
| Specificity | 0.9219 |
| Precision | 0.9219 |
| Negative Predictive Value | 0.9314 |
| False Positive Rate | 0.0331 |
| False Discovery Rate | 0.0321 |
| False Negative Rate | 0.0343 |
| **Accuracy** | **0.9556** |
| F1 Score | 0.9356 |
| Matthews Correlation Coefficient | 0.9313 |

The proposed method presented 0.9556 % which is best than famous studies proposed in this field. The comparison of our method with several studies presented in Table 4.2.

**Table 4.3:** Accuracy comparisons with previous studies

| References | Accuracy (%) |
|---|---|
| [56] | 93.52 |
| [57] | 81.16 |
| [58] | 83.16 |
| Proposed method | 95.56 |

In Table 4.3, several studies listed and analysed to compare with our method. In [56] several techniques presented to detect IDS such as DT , LR, NB, ANN, and Ramp-KSVCR. Where DT presented 85.56%, DT presented 83.15%, NB presented 82.07%, ANN presented 81.34% and Ramp-KSVCR presented 93.52%. Furthermore, in [57] GA-LR method proposed and presented 81.42% accuracy. Finally, in [58] SVM presented 83.16%.

On the other hand, our method presented 95.56% accuracy which is best than all previous studies that presented in Table 4.2.

# 5. CONCLUSION

In this study, new technique proposed to IDS recognition by utilizing SVM based BOA. This is the first run through SVM based BOA used to distinguish IDS. The proposed strategy separated touchy highlights and diminish the element of highlights by utilizing BOA. This lead to deliver 95.56% exactness which is astounding outcomes when contrasted and past examinations. The expansion in utilizing web in various fields, for example, banking, internet shopping and lodging reservation lead to increment in IDS assaults as well. Accordingly, number of studies in IDS identification utilizing programmed frameworks and strategies was expanded.

unused programmed discovery strategy proposed by utilizing SVM and BOA procedures to identify IDS assaults naturally. SVM displayed palatable comes about totally different areas such as picture acknowledgment, video classification and confront location etc. In this proposition, SVM based BOA utilized to identify IDS assaults the proposed strategy displayed momentous comes about when compared with different considers.

The major conclusion from this consider is that the BOA increment precision when compared with another classical machine learning. The utilize of BOA method displayed best comes about than conventional strategies that utilized in past considers. Subsequently, the control of profound learning procedures optimized by utilizing BOA, this combination can be connected to another classification issue too.

Just as, makers are stimulated to advance when all is said in done execution of this building for progressively complex issues, for example, 3D picture getting ready and continuous robotized system. In like way, differing heuristic advancement figuring, tallying genetic counts, particle swarm streamlining, or province enhancement estimations will be used to survey SVM based SVM parameters and contrasted and the PSO Strategy as future works. It is furthermore observed that the proposed building can too be used for thorough affirmation and estimation issues, tallying movement affirmation, URL reputation, SMS spam assortment and so on.

# REFERENCES

[1] M. Schwarz, S. Weiser, D. Gruss, C. Maurice, and S. Mangard, “Malware guard extension: Using SGX to conceal cache attacks,” in Proc. 14th Int. Conf. Detection Intrusions Malware Vulnerability Assessment, 2017, pp. 3–24.

[2] F. Brasser, U. Muller, A. Dmitrienko, K. Kostiainen, S. Capkun, € and A. Sadeghi, “Software grand exposure: SGX cache attacks are practical,” in Proc. 11th USENIX Workshop Offensive Technol., 2017, Art. no. 11.

[3] S. Lee, M. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, “Inferring fine-grained control flow inside SGX enclaves with branch shadowing,” in Proc. 26th USENIX Secur. Symp., 2017, pp. 557–574.

[4] W. Wang et al.,“Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2017, pp. 2421–2434.

[5] X. Yuanzhong, C. Weidong, and P. Marcus, “Controlled-channel attacks: Deterministic side channels for untrusted operating systems,” in Proc. IEEE Symp. Security Privacy, 2015, pp. 640–656.

[6] S. Shinde, Z. L. Chua, V. Narayanan, and P. Saxena, “Preventing page faults from telling your secrets,” in Proc. 11th ACM Asia Conf. Comput. Commun. Secur., 2016, pp. 317–328.

[7] J. V. Bulck, N. Weichbrodt, R. Kapitza, F. Piessens, and R. Strackx, “Telling your secrets without page faults: Stealthy page table-based attacks on enclaved execution,” in Proc. 26th USENIX Security Symp., 2017, pp. 1041–1056.

[8] [33] A. M. Karim, O. Karal, and F. C¸ elebi, “A new automatic epilepsy serious detection method by using deep learning based on discrete wavelet transform,” no, vol. 4, pp. 15–18, 2018.

[9] R. Guanciale, H. Nemati, C. Baumann, and M. Dam, “Cache storage channels: Alias-driven attacks and verified countermeasures,” in Proc. IEEE Symp. Security Privacy, 2016, pp. 38–55.

[10] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, and S. Mangard, “ARMageddon: Cache attacks on mobile devices,” in Proc. 25th USENIX Secur. Symp., 2016, pp. 549–564.

[11] K. Ryan, “Hardware-backed heist: Extracting ECDSA keys from qualcomm’s trustzone,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2019, pp. 181–194.

[12] F. Li, C. Yan, Z. Zhu, and D. Meng, “A deep malware detection method based on general-purpose register features,” in Proc. Int. Conf. Comput. Sci., 2019, pp. 221–235.

[13] P. Okane, S. Sezer, K. Mclaughlin, and E. G. Im, “Malware detection: Program run length against detection rate,” IET Softw., vol. 8, no. 1, pp. 42–51, Feb. 2014.

[14] P. O’Kane, S. Sezer, K. McLaughlin, and E. G. Im, “SVM training phase reduction using dataset feature filtering for malware detection,” IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, pp. 500–509, Mar. 2013.

[15] M. Ozsoy, K. N. Khasawneh, C. Donovick, I. Gorelik, N. Abu-Ghazaleh , and D. Ponomarev, “Hardware-based malware detection using low-level architectural features,” IEEE Trans. Comput., vol. 65, no. 11, pp. 3332–3344, Nov. 2016.

[16] M. Qureshi, “CEASER: Mitigating conflict-based cache attacks via encrypted-address and remapping,” in Proc. 51st Annu. IEEE/ ACM Int. Symp. Microarchit., 2018, pp. 775–787.

[17] M. K. Qureshi, “New attacks and defense for encrypted-address cache,” in Proc. 46th Int. Symp. Comput. Archit., 2019, pp. 360–371.

[18] Z. Wang and R. B. Lee, “A novel cache architecture with enhanced performance and security,” in Proc. 41st Annu. IEEE/ACM Int. Symp. Microarchit., 2008, pp. 83–93.

[19] N. Binkertetal, “The gem5 simulator,” ACM SIGARCH Comput. Archit. News, vol. 39, no. 2, pp. 1–7, 2011.

[20] W. Song and P. Liu, “Dynamically finding minimal eviction sets can be quicker than you think for side-channel attacks against the LLC,” in Proc. 22nd Int. Symp. Res. Attacks Intrusions Defenses, 2019, pp. 427–442.

[21] A. M. Karim, M. S. Guzel, M. R. Tolun, H. Kaya, and F. V. C¸ elebi, “A new framework using deep auto-encoder and energy spectral density for medical waveform data classification and processing,” Biocybernetics and Biomedical Engineering, vol. 39, no. 1, pp. 148–159, 2019.

[22] AVwares. 2019. [Online]. Available: http://avaware.com

[23] L. McVoy and C. Staelin, “lmbench: Portable tools for performance analysis,” in Proc. USENIX Annu. Tech. Conf., 1996, pp. 279–294.

[24] T. Alves and D. Felton, “TrustZone: Integrated hardware and software security-enabling trusted computing in embedded systems (July 2004),” 2014. [Online]. Available: http://docplayer.net/ 51242536-Trustzone-integrated-hardware-and-software-securityenabling-trusted-computing-in-embedded-systems.html

[25] G. Chen, S. Chen, Y. Xiao, Y. Zhang, Z. Lin, and T. H. Lai, “SgxPectre: Stealing Intel secrets from SGX enclaves via speculative execution,” in Proc. IEEE Eur. Symp. Secur. Privacy, 2019, pp. 142–157.

[26] M. Schwarz, S. Weiser, and D. Gruss, “Practical enclave malware with intel SGX,” CoRR, vol. abs/1902.03256, 2019. [Online]. Available: http://arxiv.org/abs/1902.03256

[27] M. Lipp, D. Gruss, R. Spreitzer, C. Maurice, and S. Mangard, “ARMageddon: Cache attacks on mobile devices,” in Proc. 25th USENIX Secur. Symp., 2016, pp. 549–564.

[28] V. Costan, I. Lebedev, and S. Devadas, “Sanctum: Minimal hardware extensions for strong software isolation,” in Proc. 25th USENIX Secur. Symp., 2016, pp. 857–874.

[29] T. Bourgeat, I. Lebedev, A. Wright, S. Zhang, Arvind, and S. Devadas, “MI6: Secure enclaves in a speculative out-of-order processor,” in Proc. 52nd Annu. IEEE/ACM Int. Symp. Microarchit., 2019, pp. 42–56.

[30] O. Hamza and K. Omer, “IRONHIDE: A secure multicore that efficiently mitigates microarchitecture state attacks for interactive applications,” in Proc. IEEE Int. Symp. High Perform. Comput. Archit., 2020, pp. 111–122.

[31] A.M. Karim Kaya, H.; Güzel, M.S.; Tolun, M.R.; Çelebi, F.V.; Mishra, A. A Novel Framework Using Deep Auto-Encoders Based Linear Model for Data Classification. Sensors 2020, 20, 6378.

[32] X. Ruan, Platform Embedded Security Technology Revealed: Safeguarding the Future of Computing With Intel Embedded Security and Management Engine, New York, NY, USA: Apress, 2014.

[33] C. Demerjian, “Remote security exploit in all 2008+ Intel platforms,” 2017. [Online]. Available: https://semiaccurate.com/2017/05/01/ remote-security-exploit-2008-intel-platforms/

[34] R. Millman, “Security issue found in AMD’s Platform Security Processor,” 2018. [Online]. Available: https://www.scmagazineuk. com/security-issue-found-amds-platform-security-processor/ article/1473518

[35] M. Dan et al., “Security-first architecture: Deploying physically isolated active security processors for safeguarding the future of computing,” Cybersecurity, vol. 1, 2018, Art. no. 2.

[36] N. Christoulakis, G. Christou, E. Athanasopoulos, and S. Ioannidis, “HCFI: Hardware-enforced control-flow integrity,” in Proc. 6th ACM Conf. Data Appl. Secur. Privacy, 2016, pp. 38–49.

[37] R. De Clercq , J. Gotzfried, D. € Ubler, P. Maene, and I. Verbauwhede, € “SOFIA: Software and control flow integrity architecture,” Comput. Secur., vol. 68, pp. 16–35, 2017.

[38] R. Roemer, E. Buchanan, H. Shacham, and S. Savage, “Returnoriented programming: Systems, languages, and applications,” ACM Trans. Inf. Syst. Secur., vol. 15, no. 1, 2012, Art. no. 2.

[39] T. Bletsch, X. Jiang, V. W. Freeh, and Z. Liang, “Jump-oriented programming: A new class of code-reuse attack,” in Proc. 6th ACM Symp. Inf. Comput. Commun. Secur., 2011, pp. 30–40.

[40] V. Kiriansky, I. Lebedev, S. Amarasinghe, S. Devadas, and J. Emer, “DAWG: A defense against cache timing attacks in speculative execution processors,” in Proc. 51st Annu. IEEE/ACM Int. Symp. Microarchit., 2018, pp. 974–987.

[41] F. Liu et al.,“CATalyst: Defeating last-level cache side channel attacks in cloud computing,” in Proc. IEEE Int. Symp. High Perform. Comput. Archit., 2016, pp. 406–418.

[42] P. Li, L. Zhao, R. Hou, L. Zhang, and D. Meng, “Conditional speculation: An effective approach to safeguard out-of-order execution against spectre attacks,” in Proc. IEEE Int. Symp. High Perform. Comput. Archit., 2019, pp. 264–276.

[43] M. Yan, J. Choi, D. Skarlatos, A. Morrison, C. W. Fletcher, and J. Torrellas, “InvisiSpec: Making speculative execution invisible in the cache hierarchy,” in Proc. 51th Int. Symp. Microarchit., 2018, pp. 428–441.

[44] K. N. Khasawneh, E. M. Koruyeh, C. Song, D. Evtyushkin, D. Ponomarev, and N. Abu-Ghazaleh , “SafeSpec: Banishing the spectre of a meltdown with leakage-free speculation,” in Proc. 56th Annu. Des. Automat. Conf., 2019, pp. 60:1–60:6.

[45] O. Weisse, I. Neal, K. Loughlin, T. F. Wenisch, and B. Kasikci, “NDA: Preventing speculative execution attacks at their source,” in Proc. 52nd Annu. IEEE/ACM Int. Symp. Microarchit., 2019, pp. 572–586.

[46] B. Balamurugan and D. Biswas, “Security in network layer of IoT: Possible measures to preclude,” in Security Breaches and Threat Prevention in the Internet of Things. Hoboken, NJ, USA: IGI Glob., 2017, pp. 46–75.

[47] A. M. Karim, F. V. C¸ elebi, and A. S. Mohammed, “Sofware Development for Blood Disease Expert System,” Lecture Notes on Empirical Sofware Engineering, vol. 4, no. 3, pp. 179–183, 2016.

[48] O. Arias, J. Wurm, K. Hoang, and Y. Jin, “Privacy and security in Internet of Things and wearable devices,” IEEE Trans. Multi-Scale Comput. Syst., vol. 1, no. 2, pp. 99–109, Apr.–Jun. 2015.

[49] B. Fowler, “Some top baby monitors lack basic security features report finds,” 2015.

[50] T. Borgohain, U. Kumar, and S. Sanyal. (2015). Survey of Security and Privacy Issues of Internet of Things. [Online]. Available: http://arxiv.org/abs/1501.02211

[51] V. B. Misic, J. Fang, and J. Misic, “MAC layer security of 802.15.4- compliant networks,” in Proc. IEEE Int. Conf. Mobile Adhoc Sensor Syst. Conf., 2005, p. 8.

[52] N. Sastry and D. Wagner, “Security considerations for IEEE 802.15.4 networks,” in Proc. 3rd ACM Workshop Wireless Security, 2004, pp. 32–42.

[53] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, “A dynamic prime number based efficient security mechanism for big sensing data streams,” J. Comput. Syst. Sci., vol. 83, no. 1, pp. 22–42, 2017.

[54] Karim, A. M., Güzel, M. S., Tolun, M. R., Kaya, H., and Çelebi, F. V., “A New Generalized Deep Learning Framework Combining Sparse Auto-encoder and Taguchi Method for Novel Data Classification and Processing,” pp. 1–22.

[55] R. M. Savola, H. Abie, and M. Sihvonen, “Towards metrics driven adaptive security management in eHealth IoT applications,” in Proc. 7th Int. Conf. Body Area Netw., 2012, pp. 276–281.

[56] A. Kanuparthi, R. Karri, and S. Addepalli, “Hardware and embedded security in the context of Internet of Things,” in Proc. ACM Workshop Security Privacy Dependability Cyber Veh., 2013, pp. 61–64.

[57] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, “Security and privacy challenges in industrial Internet of Things,” in Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC), 2015, pp. 1–6.

[58] Senirkentli, G.B.; Ekinci, F.; Bostanci, E.; Güzel, M.S.; Dağli, Ö.; Karim, A.M.; Mishra, A. Proton Therapy for Mandibula Plate Phantom. *Healthcare* **2021**, *9*, 167. https://doi.org/10.3390/healthcare9020167.

[1] B. Suresh Kumar, B.V.V. Siva Prasad, Sonali Vyas, Combining the OGA with IDS to improve the detection rate, Materials Today: Proceedings, 2020, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2020.09.540..

[2] Gorby Kabasele Ndonda, Ramin Sadre, Network trace generation for flow-based IDS evaluation in control and automation systems, International Journal of Critical Infrastructure Protection, Volume 31, 2020, 100385, ISSN 1874-5482, https://doi.org/10.1016/j.ijcip.2020.100385.

[3] Shuokang Huang, Kai Lei, IGAN-IDS: An imbalanced generative adversarial network towards intrusion detection system in ad-hoc networks, Ad Hoc Networks, Volume 105, 2020, 102177, ISSN 1570-8705, https://doi.org/10.1016/j.adhoc.2020.102177.

[4] Punam Bedi, Neha Gupta, Vinita Jindal, Siam-IDS: Handling class imbalance problem in Intrusion Detection Systems using Siamese Neural Network, Procedia Computer Science, Volume 171, 2020, Pages 780-789, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.04.085.

[5] Ningthoujam Chidananda Singh, Avinash Sharma, Resilience of mobile ad hoc networks to security attacks and optimization of routing process, Materials Today: Proceedings, 2020, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2020.09.622.

[6] Alberto Urueña López, Fernando Mateo, Julio Navío-Marco, José María Martínez-Martínez, Juan Gómez-Sanchís, Joan Vila-Francés, Antonio José Serrano-López, Analysis of computer user behavior, security incidents and fraud using Self-Organizing Maps, Computers & Security, Volume 83, 2019, Pages 38-51, ISSN 0167-4048,

https://doi.org/10.1016/j.cose.2019.01.009.

[7] Mohamed Amine Rguibi, Najem Moussa, Hybrid Trust Model for Worm Mitigation in P2P Networks, Journal of Information Security and Applications, Volume 43, 2018, Pages 21-36, ISSN 2214-2126, https://doi.org/10.1016/j.jisa.2018.10.005.

[8] Yiping Guo, Credit risk assessment of P2P lending platform towards big data based on BP neural network, Journal of Visual Communication and Image Representation, Volume 71, 2020, 102730, ISSN 1047-3203, https://doi.org/10.1016/j.jvcir.2019.102730.

[9] Massimo Cafaro, Italo Epicoco, Marco Pulimeno, Mining frequent items in unstructured P2P networks, Future Generation Computer Systems, Volume 95, 2019, Pages 1-16, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.12.030.

[10] Congjun Rao, Ming Liu, Mark Goh, Jianghui Wen, 2-stage modified random forest model for credit risk assessment of P2P network lending to "Three Rurals" borrowers, Applied Soft Computing, Volume 95, 2020, 106570, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2020.106570.

[11] Hui-Kai Su, Jian-Ting Pan, Kim-Joan Chen, Marek R. Ogiela, Hsing-Chung Chen, Real-Time Streaming Relay Mechanism for P2P Conferences on Hierarchical Overlay Networksb, Journal of Electronic Science and Technology, Volume 17, Issue 3, 2019, Pages 242-251, ISSN 1674-862X, https://doi.org/10.11989/JEST.1674-862X.71018158.

[12] M. Imran Azim, Wayes Tushar, Tapan K. Saha, Investigating the impact of P2P trading on power losses in grid-connected networks with prosumers, Applied Energy, Volume 263, 2020, 114687, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2020.114687.

[13] Mark Jenkins, Ivana Kockar, Impact of P2P trading on distributed generation curtailment in constrained distribution networks, Electric Power Systems Research, Volume 189, 2020, 106666, ISSN 0378-7796, https://doi.org/10.1016/j.epsr.2020.106666.

[14] Rasool Azimi, Hedieh Sajedi, Mohadeseh Ghayekhloo, A distributed data clustering algorithm in P2P networks, Applied Soft Computing, Volume 51, 2017, Pages 147-167,

ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2016.11.045.

[15] Morteza Babazadeh Shareh, Hamidreza Navidi, Hamid Haj Seyyed Javadi, Mehdi HosseinZadeh, Preventing Sybil attacks in P2P file sharing networks based on the evolutionary game model, Information Sciences, Volume 470, 2019, Pages 94-108, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2018.08.054.

[16] Sateesh Kumar Awasthi, Yatindra Nath Singh, Simplified Biased Contribution Index (SBCI): A mechanism to make P2P network fair and efficient for resource sharing, Journal of Parallel and Distributed Computing, Volume 124, 2019, Pages 106-118, ISSN 0743-7315, https://doi.org/10.1016/j.jpdc.2018.10.002..

[17] Jie Wan, Heng Zhang, Xiaoqian Zhu, Xiaolei Sun, Gang Li, Research on Influencing Factors of P2P Network Loan Prepayment Risk Based on Cox Proportional Hazards, Procedia Computer Science, Volume 162, 2019, Pages 842-848, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.12.058.

[18] Abbas Yazdinejadna, Reza M. Parizi, Ali Dehghantanha, Mohammad S. Khan, A kangaroo-based intrusion detection system on software-defined networks, Computer Networks, 2020, 107688, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2020.107688.

[19] Malu Zhang, Jibin Wu, Ammar Belatreche, Zihan Pan, Xiurui Xie, Yansong Chua, Guoqi Li, Hong Qu, Haizhou Li, Supervised learning in spiking neural networks with synaptic delay-weight plasticity, Neurocomputing, Volume 409, 2020, Pages 103-118, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2020.03.079.

[20] Jiangjiao Duan, Banghui Luo, Jianping Zeng, Semi-supervised learning with generative model for sentiment classification of stock messages, Expert Systems with Applications, Volume 158, 2020, 113540, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113540.

[21] Takahiko Furuya, Ryutarou Ohbuchi, Transcoding across 3D shape representations for unsupervised learning of 3D shape feature, Pattern Recognition Letters, Volume 138, 2020, Pages 146-154, ISSN 0167-8655, ttps://doi.org/10.1016/j.patrec.2020.07.012.

[22] Leighton M. Watson, Using unsupervised machine learning to identify changes in eruptive behavior at Mount Etna, Italy, Journal of Volcanology and Geothermal Research, Volume 405, 2020, 107042, ISSN 0377-0273, https://doi.org/10.1016/j.jvolgeores.2020.107042.

[23] Zhijie Li, Nick Rothbart, Xiaojiao Deng, Hua Geng, Xiaoping Zheng, Philipp Neumaier, Heinz-Wilhelm Hübers, Qualitative and quantitative analysis of terahertz gas-phase spectroscopy using independent component analysis, Chemometrics and Intelligent Laboratory Systems, Volume 206, 2020, 104129, ISSN 0169-7439, https://doi.org/10.1016/j.chemolab.2020.104129.

[24] Enrique Cerrillo-Cuenca, Marcela Sepúlveda, Zaray Guerrero-Bueno, Independent component analysis (ICA): A statistical approach to the analysis of superimposed rock paintings, Journal of Archaeological Science, Volume 125, 2021, 105269, ISSN 0305-4403, https://doi.org/10.1016/j.jas.2020.105269.

[25] Honggui Li, Dimitri Galayko, Maria Trocan, Multi-level adaptive neuro-fuzzy inference system-based reconstruction of 1D ISOMAP representations, Fuzzy Sets and Systems, 2020, ISSN 0165-0114, https://doi.org/10.1016/j.fss.2020.11.002.

[26] Qingchao Liu, Yingfeng Cai, Haobin Jiang, Jian Lu, Long Chen, Traffic state prediction using ISOMAP manifold learning, Physica A: Statistical Mechanics and its Applications, Volume 506, 2018, Pages 532-541, ISSN 0378-4371, https://doi.org/10.1016/j.physa.2018.04.031.

[27] Xiaowan Chen, Zhenyu Dong, Jinbao Liu, Huanyuan Wang, Yang Zhang, Tianqing Chen, Yichun Du, Li Shao, Jiancang Xie, Hyperspectral characteristics and quantitative analysis of leaf chlorophyll by reflectance spectroscopy based on a genetic algorithm in combination with partial least squares regression, Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, Volume 243, 2020, 118786, ISSN 1386-1425, https://doi.org/10.1016/j.saa.2020.118786.

[28] Chuanfang Zhang, Kaixiang Peng, Jie Dong, An extensible quality-related fault isolation framework based on dual broad partial least squares with application to the hot rolling process, Expert Systems with Applications, 2020, 114166, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.114166.

[29] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16-28.

[30] Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2018). A Taxonomy and Survey of Intrusion Detection System Design Techniques, Network Threats and Datasets. arXiv preprint arXiv:1806.03517.

[31] Tao Meng, Huichao Shi, Chi Wang, Bo Wu, Application of principal component analysis in measurement of flow fluctuation, Measurement, 2020, 108503, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2020.108503.

[32] Mohammad Reza Mahmoudi, Mohammad Hossein Heydari, Sultan Noman Qasem, Amirhosein Mosavi, Shahab S. Band, Principal component analysis to study the relations between the spread rates of COVID-19 in high risks countries, Alexandria Engineering Journal, 2020, ISSN 1110-0168, https://doi.org/10.1016/j.aej.2020.09.013.

[33] A. M. Karim, O. Karal, and F. C¸ elebi, "A new automatic epilepsy serious detection method by using deep learning based on discrete wavelet transform," no, vol. 4, pp. 15–18, 2018. l

[34] A. M. Karim, M. S. Guzel, M. R. Tolun, H. Kaya, and F. V. C¸ elebi, "A new framework using deep auto-encoder and energy spectral density for medical waveform data classification and processing," Biocybernetics and Biomedical Engineering, vol. 39, no. 1, pp. 148–159, 2019.

[35] A.M. Karim Kaya, H.; Güzel, M.S.; Tolun, M.R.; Çelebi, F.V.; Mishra, A. A Novel Framework Using Deep Auto-Encoders Based Linear Model for Data Classification. Sensors 2020, 20, 6378.

[36] Tongsheng Wang, Zhu Huang, Zhongguo Sun, Guang Xi, Reconstruction of natural

convection within an enclosure using deep neural network, International Journal of Heat and Mass Transfer, Volume 164, 2021, 120626, ISSN 0017-9310, https://doi.org/10.1016/j.ijheatmasstransfer.2020.120626.

[37] A. M. Karim, F. V. C¸ elebi, and A. S. Mohammed, "Sofware Development for Blood Disease Expert System," Lecture Notes on Empirical Sofware Engineering, vol. 4, no. 3, pp. 179–183, 2016.

[38] Ming-Zeng Liu, Yuan-Hai Shao, Chun-Na Li, Wei-Jie Chen, Smooth pinball loss nonparallel support vector machine for robust classification, Applied Soft Computing, 2020, 106840, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2020.106840.

[39] Jiale Huang, Tao Jin, Menglin Liang, Houlei Chen, Prediction of heat exchanger performance in cryogenic oscillating flow conditions by support vector machine, Applied Thermal Engineering, Volume 182, 2021, 116053, ISSN 1359-4311, https://doi.org/10.1016/j.applthermaleng.2020.116053.

[40] Qinghua Tao, Zhen Li, Jun Xu, Na Xie, Shuning Wang, Johan A.K. Suykens, Learning with continuous piecewise linear decision trees, Expert Systems with Applications, 2020, 114214, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.114214.

[41] Mahyat Shafapour Tehrany, Biswajeet Pradhan, Mustafa Neamah Jebur, Spatial prediction of flood susceptible areas using rule based decision tree (DT) and a novel ensemble bivariate and multivariate statistical models in GIS, Journal of Hydrology, Volume 504, 2013, Pages 69-79, ISSN 0022-1694, https://doi.org/10.1016/j.jhydrol.2013.09.034.

[42] Hui Zhang, Chun-Tao Liu, Jun Mao, Chen Shen, Rui-Ling Xie, Bo Mu, Development of novel in silico prediction model for drug-induced ototoxicity by using naïve Bayes classifier approach, Toxicology in Vitro, Volume 65, 2020, 104812, ISSN 0887-2333, https://doi.org/10.1016/j.tiv.2020.104812.

[43] Yogendra Narayan, Comparative analysis of SVM and Naive Bayes classifier for the SEMG signal classification, Materials Today: Proceedings, 2020, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2020.09.093.

[44] Juan Hu, Hong Peng, Jun Wang, Wenping Yu, kNN-P: A kNN classifier optimized by P systems, Theoretical Computer Science, Volume 817, 2020, Pages 55-65, ISSN 0304-3975, https://doi.org/10.1016/j.tcs.2020.01.001.

[45] Yogendra Narayan, SEMG signal classification using KNN classifier with FD and TFD features, Materials Today: Proceedings, 2020, ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2020.09.089.

[46] Ting Wang, Jaroslaw Knap, Stochastic gradient descent for semilinear elliptic equations with uncertainties, Journal of Computational Physics, 2020, 109945, ISSN 0021-9991, https://doi.org/10.1016/j.jcp.2020.109945.

[47] Wenrui Hao, A gradient descent method for solving a system of nonlinear equations, Applied Mathematics Letters, Volume 112, 2021, 106739, ISSN 0893-9659, https://doi.org/10.1016/j.aml.2020.106739.

[48] Avinash Kumar Tiwary, Suman Ghosh, Rashmi Singh, Dipti Prasad Mukherjee, B. Uma Shankar, Pratik Swarup Dash, Automated coal petrography using random forest, International Journal of Coal Geology, Volume 232, 2020, 103629, ISSN 0166-5162,https://doi.org/10.1016/j.coal.2020.103629.

[49] Anestis Antoniadis, Sophie Lambert-Lacroix, Jean-Michel Poggi, Random forests for global sensitivity analysis: A selective review, Reliability Engineering & System Safety, Volume 206, 2021, 107312, ISSN 0951-8320, https://doi.org/10.1016/j.ress.2020.107312 /

[50] Yan-Ru Guo, Yan-Qin Bai, Chun-Na Li, Yuan-Hai Shao, Ya-Fen Ye, Cheng-zi Jiang, Reverse nearest neighbors Bhattacharyya bound linear discriminant analysis for multimodal classification,Engineering Applications of Artificial Intelligence, Volume 97, 2021, 104033, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2020.104033.

[51] Saroj S. Shivagunde, Ashwani Nadapana, V. Vijaya Saradhi, Multi-view Incremental Discriminant Analysis, Information Fusion, Volume 68, 2021, Pages 149-160, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2020.10.021.

[52] Yunpeng Wang, A.W. Kandeal, Ahmed Swidan, Swellam W. Sharshir, Gamal B.

Abdelaziz, M.A. Halim, A.E. Kabeel, Nuo Yang, Prediction of tubular solar still performance by machine learning integrated with Bayesian optimization algorithm, Applied Thermal Engineering, 2020, 116233, ISSN 1359-4311, https://doi.org/10.1016/j.applthermaleng.2020.116233.

[53] Qianjin Lin, Junmei Zheng, Chun Zou, Jia Cheng, Jiarui Li, Wenxiang Xia, Haiyang Shi, An improved 3-pentanone high temperature kinetic model using Bayesian optimization algorithm based on ignition delay times, flame speeds and species profiles, Fuel, Volume 279, 2020, 118540, ISSN 0016-2361, https://doi.org/10.1016/j.fuel.2020.118540.

[54] M. Todd Young, Jacob D. Hinkle, Ramakrishnan Kannan, Arvind Ramanathan, Distributed Bayesian optimization of deep reinforcement learning algorithms, Journal of Parallel and Distributed Computing, Volume 139, 2020, Pages 43-52, ISSN 0743-7315, https://doi.org/10.1016/j.jpdc.2019.07.008.

[55] Pranab K. Muhuri, Sajib K. Biswas, Bayesian optimization algorithm for multi-objective scheduling of time and precedence constrained tasks in heterogeneous multiprocessor systems, Applied Soft Computing, Volume 92, 2020, 106274, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2020.106274.

[56] S. M. Hosseini Bamakan, H. Wang, and Y. Shi, “Ramp loss KSupport Vector Classifcation-Regression; a robust and sparse multi-class approach to the intrusion detection problem,” Knowledge-Based Systems, vol. 126, pp. 113–126, 2017.

[57] C. Khammassi and S. Krichen, “A GA-LR wrapper approach for feature selection in network intrusion detection,” Computers & Security, vol. 70, pp. 255–277, 2017.

[58] Karim, A. M., Güzel, M. S., Tolun, M. R., Kaya, H., and Çelebi, F. V., “A New Generalized Deep Learning Framework Combining Sparse Auto-encoder and Taguchi Method for Novel Data Classification and Processing,” pp. 1–22.