الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي ٠08 ماي 5491٠ بسيدي بلعباس
École Supérieure en Informatique
-08 Mai 1945- Sidi Bel Abbès

MEMOIRE

Pour En Vue de l'obtention du diplôme de **Master**
Filière:: **Informatique**
Spécialité: **Système d'Information et Web (SIW)**

Thème

Anomaly Detection using AutoEncoders: The advanced Persistent Threats case

---

Présenté par:
BOUDOUARA Nadjet
LAIB Oumaima

Soutenu le 04 Juillet 2022 Devant le jury composé de:

Dr. KECHAR Mohamed	President
Dr. BENABDERRAHMANE Sid Ahmed	Encadreur
Pr. BENSLIMANE Sidi Mohamed	Co-Encadreur
Dr. KHALDI Miloud	Examinateur

Année Universitaire : 2021/2022

Acknowledgments

*w*e first want to express our sincere thanks to Allah for helping us and giving us the patience and motivation to complete this work.

Our gratitude and thanks go to Our supervisors **Pr Benslimane Sidi-Mohamede** ,**Dr Benabderahmane Sid Ahmed** for guiding and advising us during the realization of this work, and for the time they devoted to us to realize this work.

We would like to thank our dear parents for giving us the help, encouragement, support and the necessary means and conditions to succeed in our university studies, without forgetting all the members of our family and our close friends.

We close by thanking all my professors and all the staff of the ESI-SBA.

Abstract

In the recent decade, there has been a massive increase in the number of apps with linked users and services for that Anomaly detection became a significant subject that has been studied in a variety of academic fields and application domains. Many anomaly detection approaches have been developed expressly for certain application areas, while others are more general.This study aims to give an organized and thorough review of anomaly detection research. We classified existing approaches into several groups depending on the underlying strategy used by each technique.

We defined essential assumptions for each category that are employed by the strategies to distinguish between normal and abnormal behavior. We propose a fundamental anomaly detection approach for each category and then illustrate how the many current techniques in that category are modifications of the basic technique. This template simplifies and condenses comprehension of the procedures in each area.

We believe that this study will give a better knowledge of the various areas in which research on this issue has been conducted, as well as how approaches created in one field can be utilized in domains for which they were not originally intended.

Keywords: Anomaly detection, Machine learning, Deep learning, cybersecurity, Advanced persistent threats, Big Data, Autoencoder.

Résumé

Au cours de la dernière décennie, il y a eu une augmentation massive du nombre d'applications avec des utilisateurs et des services liés pour que la détection d'anomalies soit devenue un sujet important qui a été étudié dans une variété de domaines académiques et de domaines d'application. De nombreuses approches de détection d'anomalies ont été développées expressément pour certains domaines d'application, tandis que d'autres sont plus générales. Cette étude vise à donner un examen organisé et approfondi de la recherche sur la détection d'anomalies. Nous avons classé les approches existantes en plusieurs groupes en fonction de la stratégie sous-jacente utilisée par chaque technique.

Nous avons défini des hypothèses essentielles pour chaque catégorie qui sont employées par les stratégies pour faire la distinction entre un comportement normal et anormal.

Nous proposons une approche fondamentale de détection des anomalies pour chaque catégorie, puis illustrons comment les nombreuses techniques actuelles de cette catégorie sont des modifications de la technique de base. Ce modèle simplifie et condense la compréhension des procédures dans chaque domaine. Nous croyons que cette étude permettra de mieux connaître les différents domaines dans lesquels la recherche sur cette question a été menée, ainsi que la façon dont les approches créées dans un domaine peuvent être utilisées dans des domaines pour lesquels elles n'étaient pas initialement destinées.

Mot Clé : Détection d'anomalies, Machine learning, Deep learning , cybersécurité, Menaces persistantes avancées, Big Data, Autoencodeur.

مُلخص

في العقد الأخير ، كانت هناك زيادة هائلة في عدد التطبيقات مع المستخدمين المرتبطين والخدمات التي أصبحت موضوعاً مهمًا تمت دراسته في مجموعة متنوعة من المجالات الأكاديمية ومجالات التطبيق. تم تطوير العديد من مناهج اكتشاف الشذوذ بشكل صريح لمجالات تطبيق معينة ، في حين أن البعض الآخر أكثر عمومية. تهدف هذه الدراسة إلى تقديم مراجعة منظمة وشاملة لأبحاث الكشف عن الشذوذ. قمنا بتصنيف الأساليب الحالية إلى عدة مجموعات اعتمادًا على الاستراتيجية الأساسية المستخدمة بواسطة كل تقنية. حددنا الافتراضات الأساسية لكل فئة التي تستخدمها الاستراتيجيات للتمييز بين السلوك الطبيعي وغير الطبيعي. نقترح نهجًا أساسيًا للكشف عن الشذوذ لكل فئة ثم نوضح كيف أن العديد من التقنيات الحالية في هذه الفئة تعد تعديلات على التقنية الأساسية. يعمل هذا القالب على تبسيط وتكثيف فهم الإجراءات في كل منطقة. بالإضافة إلى ذلك ، لكل فئة.

نعتقد أن هذه الدراسة ستوفر معرفة أفضل بالمجالات المختلفة التي أجريت فيها الأبحاث حول هذه المسألة ، وكذلك كيف يمكن استخدام الأساليب التي تم إنشاؤها في مجال واحد في المجالات التي لم تكن مخصصة لها في الأصل.

الكلمات المفتاحية : كشف الشذوذ ، التعلم الآلي ، التعلم العميق ، الأمن السيبراني ، التهديدات المستمرة المتقدمة ، البيانات الضخمة ، التشفير التلقائي.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

- **AI:** Artificial Intelligence
- **ML:** Machine Learning
- **DL:** Deep Learning
- **KNN:** K Nearest Neighbor
- **SVM:** Support Vector Machines
- **NB:** Naive Bayes
- **RF:** Random Forest
- **ANN:** Artificial Neural Networks
- **CV:** Cross Validation
- **DCG:** Discounted cumulative gain
- **NDCG:** Normalized Discounted cumulative gain
- **AUC:** Area Under the Curve
- **ROC:** Receiver Characteristic Operator
- **APT:**Advanced Persistent Threat
- **NN:**Neural Network
- **ILA:**Inductive Learning Algorithm
- **NLP:**Natural Language Processin
- **MLP:**Multi Layer Perceptron
- **PCA:**Principal Components Analysis
- **DAE:**Denoising Autoencoders
- **CAE:**Contractive Autoencoders

- **MCE:**Mean Square Error
- **MAE:**Mean Absolute Error
- **MSLE:**Mean Square Logarithmic Error
- **GD:**Gradient Descent
- **SGD:** Stochastic Gradient Descent
- **TP:** True Positives
- **TN:** True Negatives
- **FP:** False Positive
- **FN:** False Negative
- **MDL :** Minimum description length

Part I

Introduction

CHAPTER 1

INTRODUCTION

1.1 Introduction :

Cyber attacks have existed since the Internet's birth and have developed significantly over the years, from viruses and worms within the period to malware , botnets .

Recently within the connected digital world, targeted attack has become one amongst the foremost serious threats to standard computing systems. Advanced persistent threat (APT) is currently one in every of the foremost important threats considering the knowledge security concept .APT collects data from a specified target on a continuous basis by exploiting vulnerabilities with a variety of attack methodologies. Many researchers have contributed to the development of methodologies and solutions to combat network intrusion and malicious malware. However, only a few of these solutions are specifically targeted towards APT.APT attackers may employ common ways to get access to their target entity's network, but the tools they use to do so are unfamiliar. The tools utilized are sophisticated, as the word implies, and they must be in order for an attacker to remain persistent in the network for longer periods of time. They stay low, gradually increasing their footing from one system to another within the organization's network, gathering important information as they travel and strategically exporting it to their command and control center. APTs are often carried out by well-funded attackers who are given the resources they require to carry out the assault for as long as the funding organization requires.

The attack only stops when it is detected or when the financing organization has all of the necessary data ,affected organizations of APT attacks are put into question about their failure to notice the assault despite having security measures such as powerful intrusion detection and prevention systems.

This thesis provides readers with an overview of anomaly detection for The advanced Persistent Threats case approaches using machine learning and deep learning techniques. In addition, a comparison between those approaches is included.

Part II

Background Chapter

CHAPTER 2

ADVANCED PERSISTENT THREATS (APTS)

According to the Gartner report, sophisticated hacking attacks are continuously increasing in the cyber space. Hacking in the past leaked personal information or were done for just fame, but recent hacking targets companies, government agencies. This kind of attack is commonly called APT(Advanced Persistent Threat).So what is APTs and what are the risks involved. We will learn about this and more in this chapter.

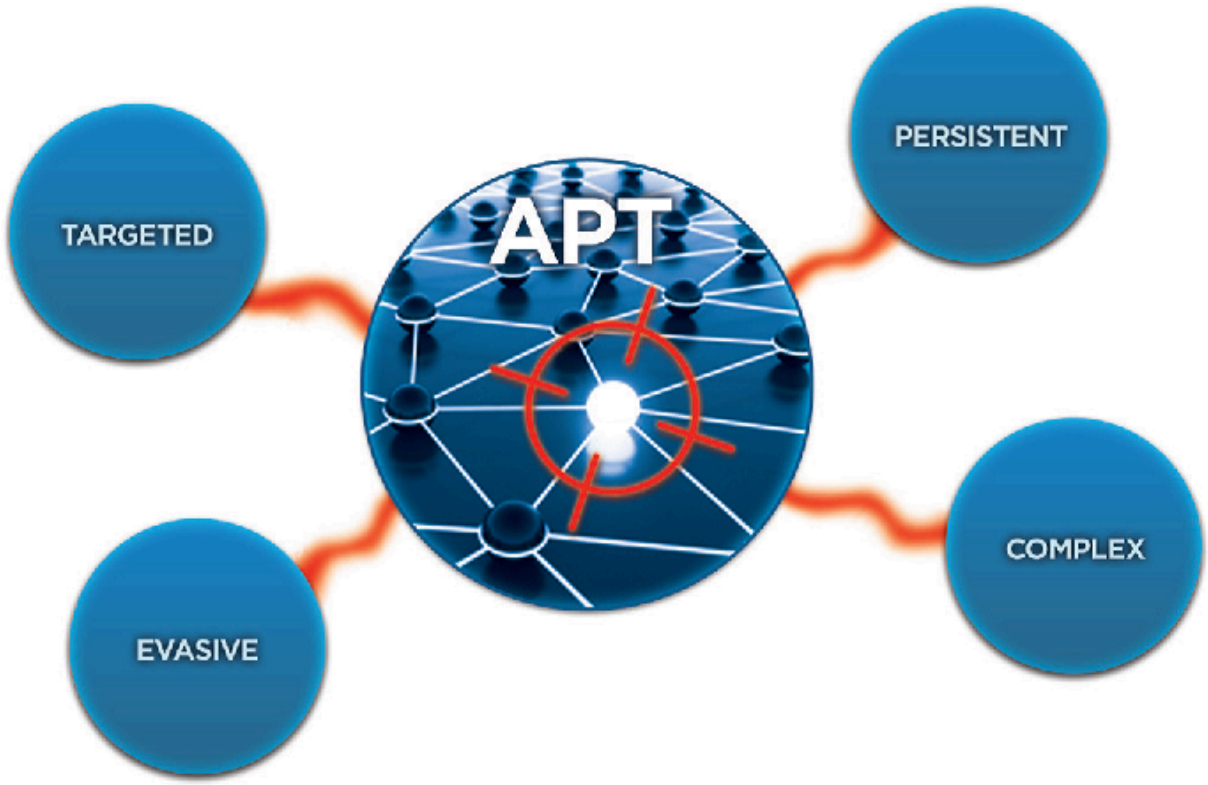

Figure 2.1: Advanced Persistent Threat [1]

[1] https://www.semanticscholar.org/paper/Advanced-Persistent-Threat-(-APT-)-Beyond-the-hype-Ask

2.1 Basic Concepts of APTs

2.1.1 What is APTs :

The term Advanced Persistent Threat refers to a well-coordinated group of people with evil purpose who attack a certain organization, government, or corporation (usually controlled by an organized body such as a government, terrorist group, corporate entity, or other(Vukalović and Delija 2015a).The APT life cycle is shown in Figure 1.2(Sneha Sakhareć 2020) .

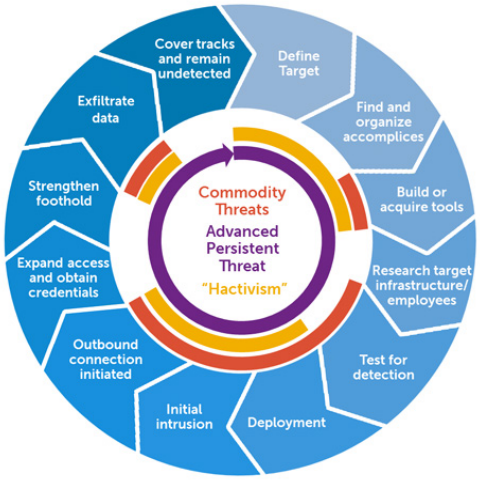

Figure 2.2: APT Life Cycle (Sneha Sakhareć 2020) .

-Initial compromise: Using social engineering and spear phishing, as well as zero-day infections, the attack was carried out by email. Planting malware on a website that the affected workers are likely to visit is another prevalent attack strategy.

-Establish Foothold: Install remote administration software on the victim's network, as well as network backdoors and tunnels that allow for covert access to the victim's infrastructure.

-Escalate Privileges : Exploits and password cracking are used to get administrator access on the victim's PC, which can then be expanded to include Windows domain administrator accounts.

-Internal Reconnaissance: Gather data about the environment's architecture, trust connections, and Windows domain structure.

-Move Laterally: Extend control to more workstations, servers, and infrastructure components, and collect data from them.

-Maintains Presence: Maintain control over the access channels and credentials you obtained in the previous phases.

-Complete Mission: Ex filters stolen information from the victim's network.

2.1.2 APTs's Purpose :

APTs have a very particular target in mind, and their main goal is to get long-term access to the target's intellectual property, sensitive information, or other interesting data that the attacker may utilize. To do this, the attack must remain undetected. (Vukalović and Delija 2015b)

APT attacks most typically target computer systems, although attackers can also utilize more traditional and easier attack tactics like social engineering or phone call interception. Assaults are also carried out by APTs in order to gather data that will aid them in the execution of future attacks(Vukalović and Delija 2015b).
APT attacker goals, and consequences faced by organizations, include: [2]

- Theft of intellectual property
- Theft of classified data
- Theft of Personally Identifiable Information (PII) or other sensitive data
- Sabotage, for example database deletion
- Complete site takeover
- Obtaining data on infrastructure for reconnaissance purposes
- Obtaining credentials to critical systems
- Access to sensitive or incriminating communications

2.1.3 Real examples of APTs attacks :

APT attacks cause damage to business operations of organizations affected by them.Some of the most notable 21st century APT attacks include:

- **Titan Rain (2003):**Titan Rain was a series of coordinated attacks on computer systems in the United States since 2003; they were known to have been ongoing for at least three years. The attacks originated in Guangdong, China. The activity is believed to be associated with a state-sponsored advanced persistent threat. It was given the designation Titan Rain by the federal government of the United States[3].
- **Stuxnet (2006):**was a very intelligent malware that was developed to attack Iran's nuclear facilities and make them malfunction. It is known to be acting in the wild for a few years until it was discovered by security researchers (Ahn and Chung ć 2014).
- **APT28 (2014):**Trend Micro detected a Russian outfit known as Fancy Bear, Pawn Storm, and Sednit in 2014. Attacked military and government targets in Ukraine and Georgia, as well as NATO institutions and American defense contractors(Mwiki et al. 2019).

[2]https://www.cynet.com/advanced-persistent-threat-apt-attacks/
[3]https://en.wikipedia.org/wiki/Titan$_R ain$

- **Deep Panda (2015):**An APT assault against the Office of Personnel Management of the United States Government, most likely launched from China. In 2015, a well-known assault code-named Deep Panda exposed over 4 million US personnel data, perhaps including information regarding secret service employees [4].
- **APT34 (2017):**In 2017, FireEye researchers discovered a group linked to Iran. It targeted Middle Eastern government organizations as well as banking, oil, chemical, and telecommunications firms [5].

2.2 Conclusion :

Hacking was formerly used to disclose personal information, but currently it is used to attack businesses, government organizations, and other institutions. APT is a term used to describe this type of assault (Advanced Persistent Threat). APT attacks a certain system and spends a lengthy time analyzing its weaknesses, and this is what we touched upon in the previous chapter

[4]https://www.cynet.com/advanced-persistent-threat-apt-attacks/
[5]https://www.cynet.com/advanced-persistent-threat-apt-attacks/

CHAPTER 3

ARTIFICIAL INTELLIGENCE (AI)

Nowadays, the terms ”Artificial Intelligence”, ”Machine Learning” and ”Deep Learning” have become very popular. They are sometimes used interchangeably, but are not synonyms, and the distinctions are important.
In this chapter, we will look at the definitions of Artificial Intelligence, Machine Learning, and Deep Learning, as well as the differences between them and the possible relationships between them.

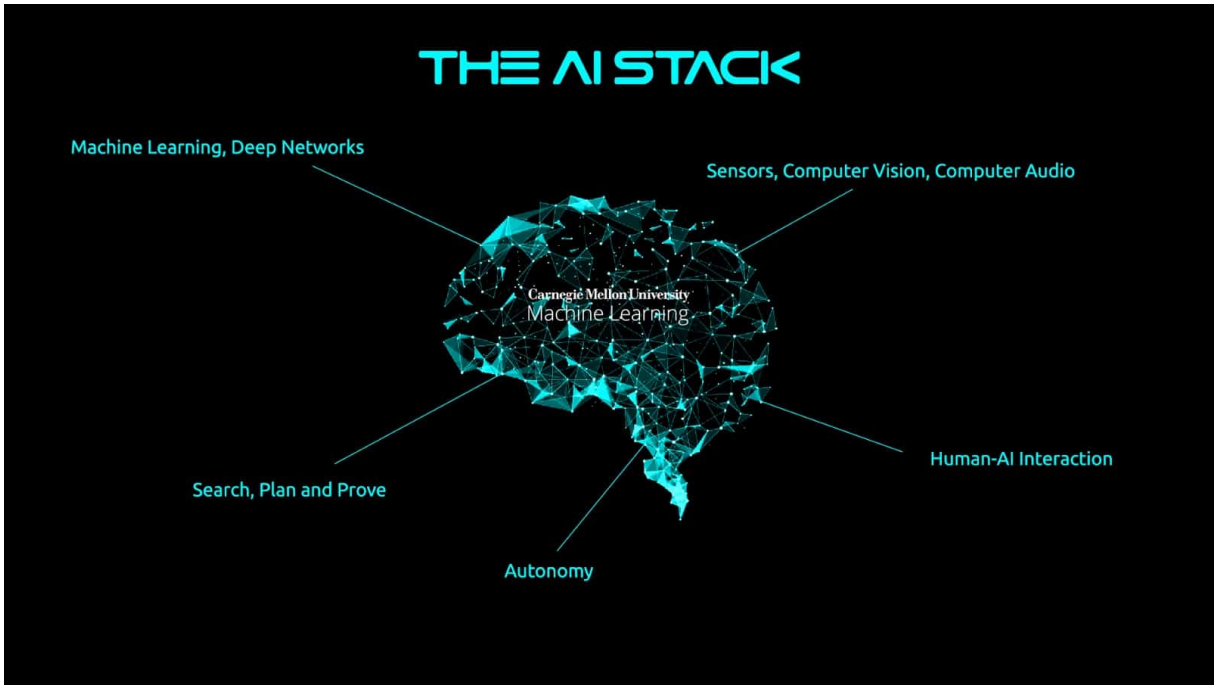

Figure 3.1: Artificial Intelligence and its main categories [1].

[1] http://ticnews.com.br/wp-content/uploads/2019/05/ML-5.jpeg

3.1 Artificial Intelligence:

3.1.1 What Is Artificial Intelligence?

Artificial intelligence (AI) is a field of science and technology that develops intelligent robots and computer programs to accomplish activities that would normally need human intellect.

3.1.2 What Is the deference between AI ,ML and DL?

"What is the difference between Artificial Intelligence (AI) and Machine Learning (ML) - and does deep learning (DL) belong to either AI or ML?" is perhaps the most often asked question. A brief official response: Machine Learning is a subset of Artificial Intelligence, while Deep Learning is a subset of Machine Learning: DL $\subset$ ML $\subset$ AI(Holzinger et al. 2018).

This follows the popular Deep Learning textbook by Ian Goodfellow, Yoshua Bengio and Aaron Courville Courville (2016, see Fig. 1):

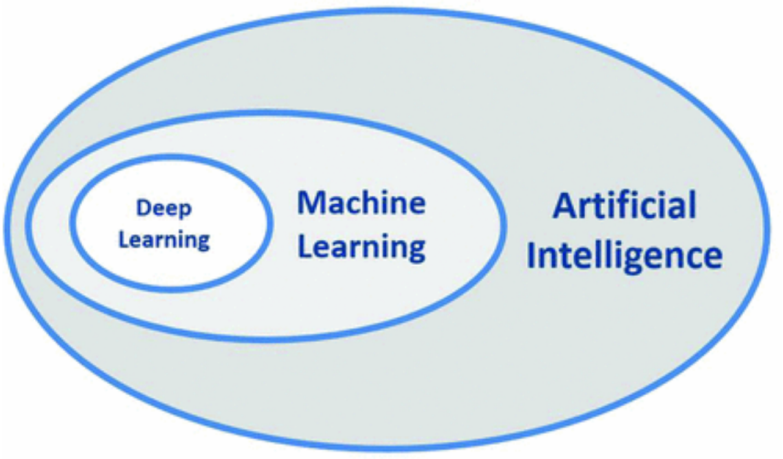

Figure 3.2: Relationship between AI - ML - DL (Ian Goodfellow 2016)

3.2 Machine Learning :

3.2.1 What Is Machine Learning?

Machine learning is a subfield of artificial intelligence (AI) and computer science that focuses on using data and algorithms to mimic how people learn, progressively improving its accuracy [2].

A more broad definition is as follows (Géron 2019):

"[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed." —Arthur Samuel, 1959

And here's one for the engineers:

"A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E". —Tom Mitchell, 1997

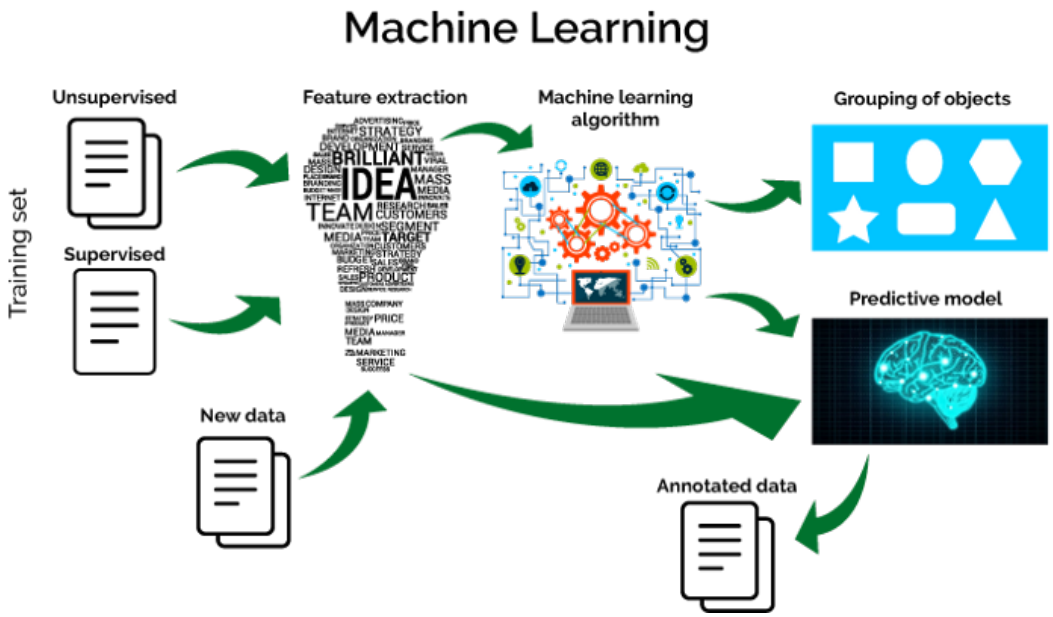

Figure 3.3: Architecture of Machine Learning

3.2.2 Machine Learning Types

there are a variety of various forms of learning, ranging from whole fields of research to individual methodologies.

In this section, you'll Know about the many forms of learning that you could face in the subject of machine learning.

In a post (Brownlee 2019) published on November 11, 2019, Jason Brownlee discussed the many forms of learning in machine learning:

1. Learning Problems :
    - Supervised Learning
    - Unsupervised Learning
    - Reinforcement Learning

[2] https://www.ibm.com/cloud/learn/machine-learning

2. Hybrid Learning Problems
    - Semi-Supervised Learning
    - Self-Supervised Learning
    - Multi-Instance Learning
3. Statistical Inference
    - Inductive Learning
    - Deductive Learning
    - Transductive Learning
4. Learning Techniques
    - Multi-Task Learning
    - Active Learning
    - Online Learning
    - Transfer Learning
    - Ensemble Learning

3.2.2.1 Supervised Learning

Supervised Learning is a machine learning method that uses a collection of paired input-output training samples to learn a system's relationship information. It's also known as Learning with a Teacher (Haykin, 1998) or Learning from Labeled Data (Haykin, 1998). (Kotsiantis, 2007). The objective of supervised learning is to create an artificial system that can learn the input-output mapping and anticipate the system's output given fresh inputs. Learning-model parameters are widely used to express information about input-output relationships(Liu and Wu 2012).

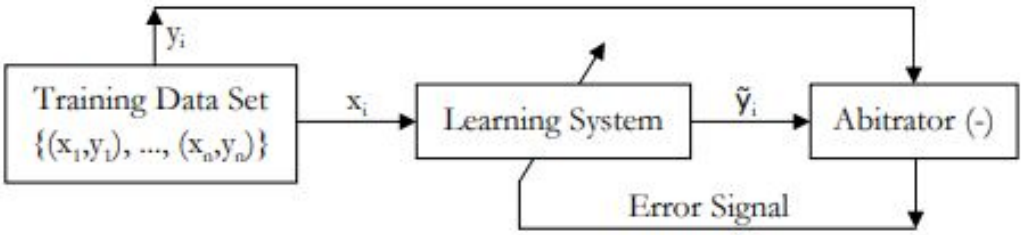

Figure 3.4: Block diagram that illustrates the form of Supervised Learning (Liu and Wu 2012)

The following are the most well-known supervised learning methods(Goyal 2021):

- Regression : Using training data, regression generates a single output value,this value is a probabilistic interpretation determined by taking into account the strength of correlation between the input variables. In logistic regression, the output has discrete values based on a set of independent variables.Furthermore, it is not adaptable enough to capture complex relationships in datasets.

- Classification : It means categorizing the information. Binary classification occurs when the supervised learning system labels incoming data into two separate classes. Multiple classifications refer to the categorization of data into more than two classes.

- Naive Bayesian Model: For big finite datasets, the Bayesian classification model is utilized. It is a method for assigning class labels that makes use of a direct acyclic network. The graph is made up of one parent node and several child nodes. Furthermore, each child node is expected to be autonomous and distinct from the parent.

- Decision Trees : A decision tree is a flowchart-like architecture that comprises conditional control statements that include decisions and their likely outcomes. The outcome is related to the labeling of unanticipated data. The leaf nodes in the tree indicate class labels, whereas the inside nodes represent characteristics. A decision tree can be used to solve issues that include both discrete qualities and boolean functions. ID3 and CART are two well-known decision tree algorithms.

- Random Forest Model : An ensemble technique is the random forest model. It works by creating a large number of decision trees and then classifying the individual trees. It may, for example, predict which undergraduate students would do well on the GMAT - a test used to get into graduate management schools.

- Neural Networks : This techniques is used to group data, identify patterns, and interpret sensory inputs. Despite their numerous benefits, neural networks need a large amount of computer power. When there are thousands of observations, fitting a neural network might get difficult. It's also known as the 'black-box' algorithm since deciphering the reasoning behind its forecasts is difficult.

- Support Vector Machines (SVM) : Is a supervised learning technique that was created in 1990. SVMs are utilized in a variety of disciplines and are strongly linked to the kernel framework. Bioinformatics, pattern recognition, and multimedia information retrieval are among examples. SVM is a discriminative classifier since it separates hyperplanes.

3.2.2.2 Unsupervised Learning

Unsupervised Learning is a machine learning approach that does not need individuals to oversee the model. Instead, it allows the model to develop on its own to identify previously unnoticed patterns and information. It mostly works with unlabeled data.

Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc (Johnson 2022).

3.2.2.2.1 Clustering Clustering is a key concept in unsupervised learning. It is primarily concerned with identifying a structure or pattern in a set of uncategorized data.

Figure 3.5: Clustering (Johnson 2022)

(*Daniel Johnson* 2022)explored the different types with Example of Unsupervised Machine Learning in a post (Johnson 2022) published on February 12, 2022:

- Exclusive (partitioning) : In this clustering method, Data are grouped in such a way that one data can belong to one cluster only.

  Example: K-means

- Agglomerative : In this clustering technique, every data is a cluster. The iterative unions between the two nearest clusters reduce the number of clusters.

  Example: Hierarchical clustering

- Overlapping : In this technique, fuzzy sets is used to cluster data. Each point may belong to two or more clusters with separate degrees of membership. Here, data will be associated with an appropriate membership value.

  Example: Fuzzy C-Means

- Probabilistic : This technique uses probability distribution to create the clusters .

  Example: Following keywords

  "man's shoe." "women's shoe." "women's glove." "man's glove." can be clustered into two categories "shoe" and "glove" or "man" and "women."

3.2.2.3 Reinforcement Learning

It's all about taking the right steps to maximize your benefit in a given circumstance. It is used by a variety of software and computers to determine the best feasible action or path in a given scenario. Reinforcement learning differs from supervised learning in that supervised learning includes the answer key, allowing the model to be trained with the correct answer, whereas reinforcement learning does not include an answer and instead relies on the reinforcement agent to decide what to do to complete the task. It is obligated to learn from its experience in the absence of a training dataset.[3]

[3]https://www.geeksforgeeks.org/what-is-reinforcement-learning/

3.2.2.4 Semi-Supervised Learning

Semi-supervised learning is a field of machine learning that focuses on performing particular learning tasks using both labeled and unlabeled data. It allows leveraging the massive volumes of unlabelled data accessible in many use cases in conjunction with generally smaller sets of labelled data. It is conceptually set between supervised and unsupervised learning(Van Engelen and Hoos 2020).

3.2.2.4.1 Taxonomy of semi-supervised learning methods These approaches differ in terms of the semi-supervised learning assumptions they utilize, how they handle unlabeled data, and how they interact with supervised algorithms. Existing classifications of semi-supervised learning techniques often employ just a subset of these qualities and are rather flat, failing to identify commonalities across distinct groups of approaches.

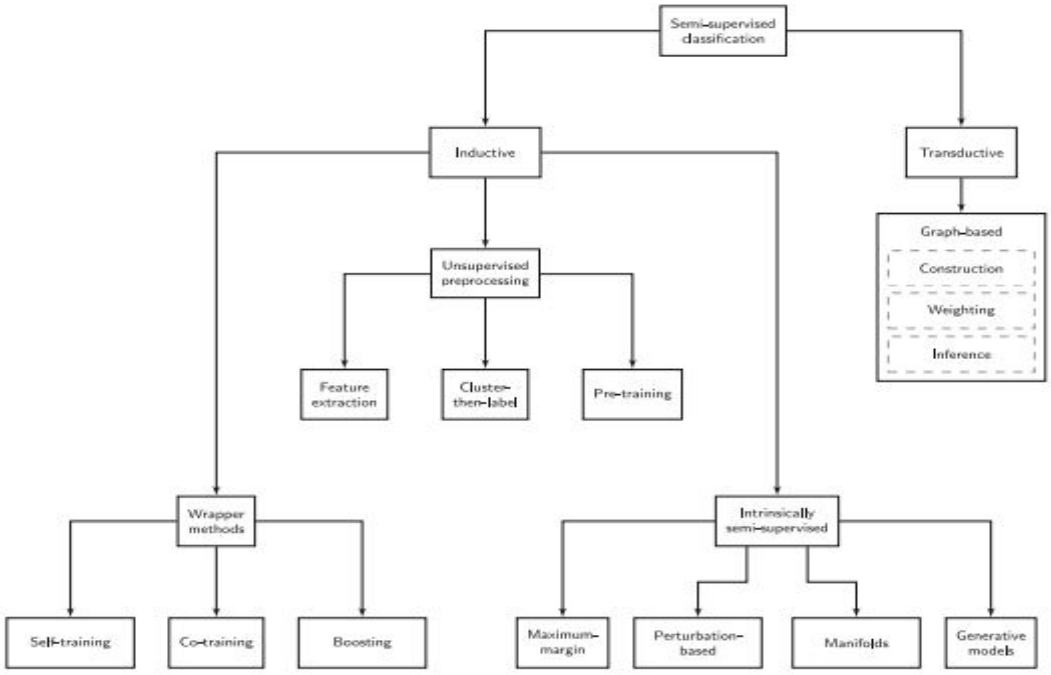

Figure 3.6: Visualization of the semi-supervised classification taxonomy(Van Engelen and Hoos 2020)

Each leaf in the taxonomy represents a distinct technique of adding unlabeled input into categorization systems. The dashed boxes in the leaves corresponding to transductive, graph-based approaches reflect various phases of the graph-based classification process, each of which has a plethora of variations.

3.2.2.5 Self-Supervised Learning

Because of its potential to avoid the cost of annotating large-scale datasets, self-supervised learning has grown in popularity. It may employ self-defined pseudolabels as supervision and use the learnt representations for a variety of downstream tasks. Contrastive learning, in particular, has lately emerged as a key component of self-supervised learning for computer vision, natural language processing (NLP), and other disciplines. It attempts to push away embeddings from distinct samples by embedding enhanced copies of the same sample next to each other(Jaiswal et al. 2021).

3.2.2.6 Multi-Instance

Individual examples are not labeled in multi-instance learning; instead, bags or groupings of samples are tagged (Brownlee 2019).

3.2.2.7 Inductive Learning

The Inductive Learning Algorithm (ILA) is an iterative and inductive machine learning algorithm that is used to generate a set of classification rules for a collection of instances, producing rules of the type "IF-THEN" at each iteration and adding to the list of rules (madarsh986 2022).

3.2.2.8 Deductive Learning

Deductive learning is a branch of machine learning that explores techniques for acquiring provably correct information. Typically, such strategies are employed to speed up problem solvers by adding information that is deductively implied by current knowledge but may result in speedier answers(Sammut and Webb 2010).

3.2.2.9 Transductive Learning

In the realm of statistical learning theory, the term "transduction" or "transductive learning" refers to predicting particular cases given specific examples from a domain (Brownlee 2019).

3.2.2.10 Multi-Task Learning

Multi-task learning is a sort of supervised learning in which a model is fitted to a single dataset to solve numerous tasks (Brownlee 2019).

3.2.2.11 Active Learning

The phrase "active learning" refers to a learning issue or system in which the learner has some control over the data used to train it. In contrast to Active Learning, where the learner is simply given a training set over which it has no control, Passive Learning involves the student being provided with a training set over which it has no influence(Cohn 2017).

3.2.2.12 Online Learning

Online learning entails using the data at hand and changing the model immediately before making a prediction or after the final observation (Brownlee 2019).

3.2.2.13 Transfer Learning

Transfer learning is a sort of learning in which a model is initially trained on one task and then utilized as the starting point for another activity (Brownlee 2019).

3.2.2.14 Ensemble Learning

Ensemble learning is a technique that involves fitting two or more models to the same data and combining the predictions from each model (Brownlee 2019).

3.3 Deep Learning :

3.3.1 What Is Deep learning?

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier(Brush 2021).

3.3.2 Neural Network:

3.3.2.1 Biologic Neural:

Neurons are excitable cells that are connected to each other and have the function of transmitting information in our nervous system. Several dendrites, a cell body, and an axon make up each neuron, as seen in the following diagram :

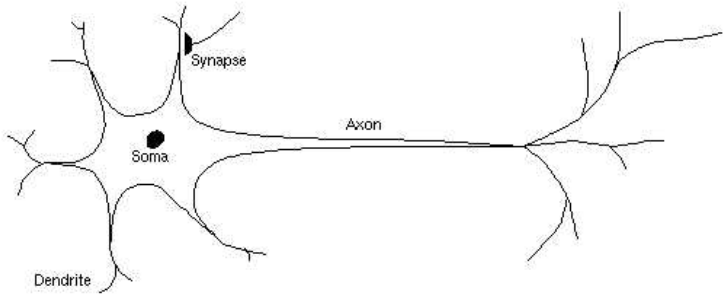

Figure 3.7: Biological Neural [4].

3.3.2.2 Artificial Neural Network

We start our examination of artificial neuron models by defining some commonly used terminology that defines the link between biological and artificial neurons . When allowed to take arbitrary nonbinary values, node output symbolizes firing frequency; however, in other artificial neural networks with binary node outputs, the analogy with biological neurons is more direct, and a node is said to be fired when its net input exceeds a particular threshold. Figure 3.8 (Mehrotra, Mohan, and Preface 1997) depicts a broad model that encompasses nearly all artificial neuron models proposed thus far. Even this ambiguous model requires the following assumptions, which raise doubts about its biological plausibility.
1. The position of the incoming synapse (connection) on the neuron (node) is immaterial (Mehrotra, Mohan, and Preface 1997).
2. Each node has a single output value, which is transmitted to all other nodes via outgoing links, regardless of their location (Mehrotra, Mohan, and Preface 1997).

[4]https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html

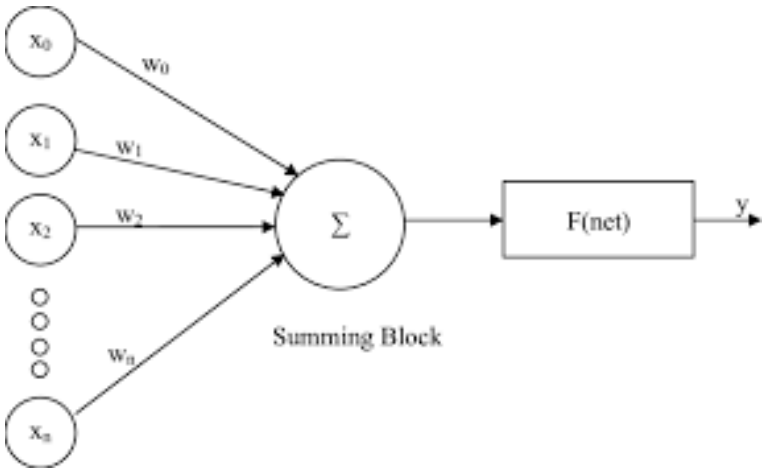

Figure 3.8: General neuron model.(Al-Jaberi 2018)

3. All inputs arrive at the same time or remain at the same degree of activation for long enough for computation to take place. Another option is to assume that buffers exist within nodes to store weighted inputs (Mehrotra, Mohan, and Preface 1997).

3.3.2.3 Perceptron Simple

A mathematical representation of a biological neuron is the perceptron. While the dendrite of real neurons receives electrical impulses from the axons of other neurons, these electrical signals are represented numerically in the perceptron. Electrical impulses are modified in varied degrees at synapses between dendrites and axons. In the perceptron, this is also approximated by multiplying each input value by a weight value. Only when the combined intensity of the input signals exceeds a specified threshold does a real neuron fire an output signal. In a perceptron, we mimic this phenomena by computing the weighted sum of the inputs, which represents the entire intensity of the input signals, and then applying a step function to the sum to produce the output.This output is supplied to other perceptrons, much like in biological neural networks ([5]).

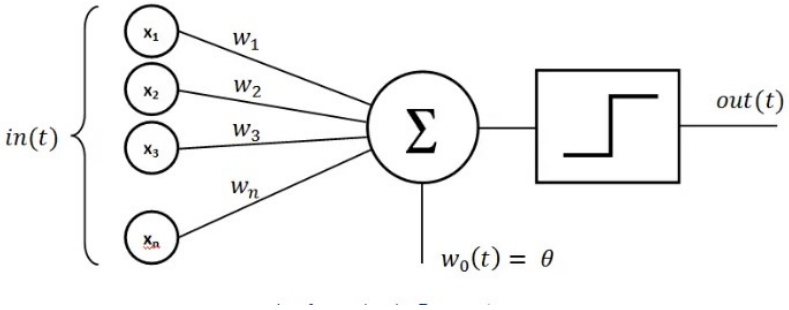

Figure 3.9: Simple Perceptron [6]

There are a number of terminology commonly used for describing neural networks. They are listed in the table below [7] :

[5]https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html
[6]https://datascientest.com/perceptron
[7]https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Neuron/index.html

The input vector	All the input values of each perceptron are collectively called the input vector of that perceptron
The weight vector	Similarly , all the weight values of each perceptron are collectively called the weight vector of that perceptron

Table 3.1: An artificial neuron (perceptron)

3.3.2.4 Multilayer Perceptron :

The multi layer perceptron (MLP) is a feed forward neural network augmentation. The input layer, output layer, and hidden layer are the three types of layers represented in Fig. 2.5. The input signal to be processed is received by the input layer. The output layer is responsible for tasks such as prediction and categorization. The real computational engine of the MLP is an arbitrary number of hidden layers inserted between the input and output layers. In an MLP, data travels from input to output layer in the forward direction, similar to a feed forward network(Abirami and Chitra 2020).

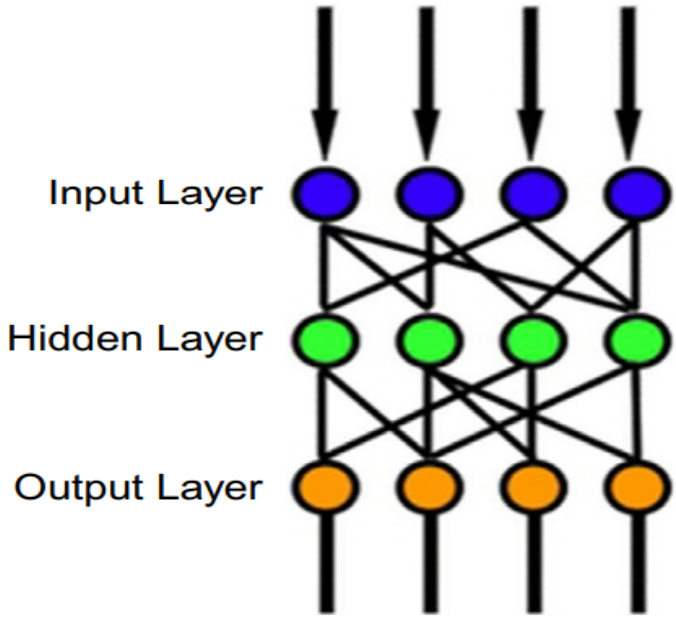

Figure 3.10: Multilayer Perceptron(Abirami and Chitra 2020)

3.3.2.5 Deep autoencoders :

As illustrated in Fig. 2.6, a deep autoencoder is a special deep learning technique that consists of two symmetrical deep belief networks with four or five shallow layers. Half of the network does the job (Abirami and Chitra 2020).

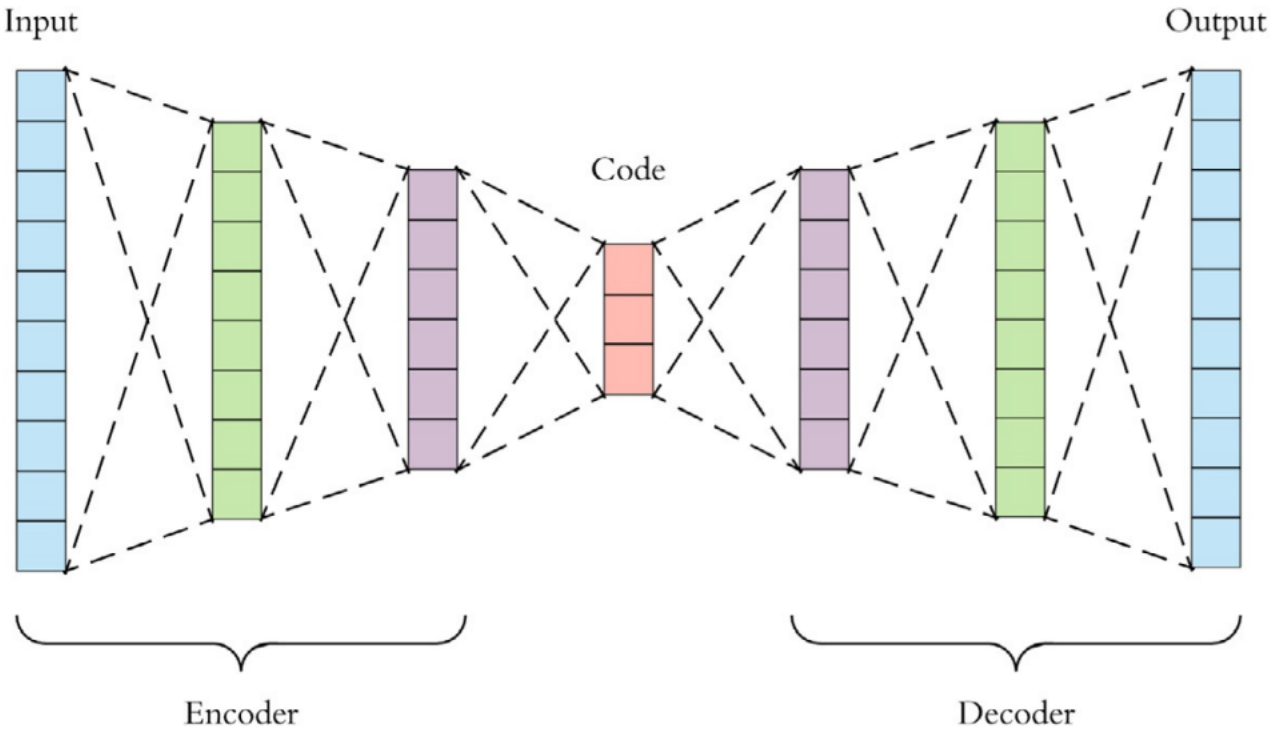

Figure 3.11: Deep autoencoders (Abirami and Chitra 2020).

The first half does the encoding, while the second half handles the decoding. The neural network family includes autoencoders. They're similar to Principal Components Analysis (PCA), but they're a lot more versatile. Autoencoders provide versatility by allowing for both linear and nonlinear encoding transformations, whereas PCA can only conduct linear transformations. By decreasing the reconstruction error between the input and output data, the autoencoders learn key characteristics existing in the data (Abirami and Chitra 2020). In autoencoders, the number of neurons in the output layer is exactly the same as the number of neurons in the input layer. The different types of Autoencoders are (Abirami and Chitra 2020) :

- Under complete Autoencoders
- Sparse Autoencoders
- Denoising Autoencoders (DAE)
- Contractive Autoencoders (CAE).

3.3.3 Activation Functions :

3.3.3.1 What is an Activation Functions ?

An Activation Function decides whether a neuron should be activated or not. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations. The role of the Activation Function is to derive output from a set of input values fed to a node (or a layer) (Sagar Sharma, Simone Sharma, and Athaiya 2017).

3.3.3.2 Types Of Activation Functions (Sagar Sharma, Simone Sharma, and Athaiya 2017):

1. **Binary Step Function :**The Binary Step Function is the most basic activation function available, and it may be written in Python using simple if-else expressions. Binary activation functions are commonly utilized while developing a binary classifier. The binary step function can be expressed mathematically as: **f(x)=1 , x** $\geqslant$ 1 and **f(x)=0 , x** $<$ **0** .

2. **Linear Activation Function :**The input is directly proportional to the linear activation function. The fundamental disadvantage of the binary step function was that it had zero gradient due to the absence of an x component. A linear function can be used to eliminate this. It can be defined as follows:**F(x)=ax.** .

3. **Sigmoid Activation Function :**Because it is a non-linear function, it is the most often utilized activation function. The sigmoid function changes data in the 0 to 1 range. It may be summed up as follows: **f(x)=1** /exp(x)**.**.

4. **Tanh Function :**The Hyperbolic Tangent function is what it's called. The tanh function resembles the sigmoid function, however it is symmetric around the origin. As a result, the outputs from previous levels will have various signs when supplied as input to the following layer.It may be summed up as follows:**f(x) = 2sigmoid(2x)-1**.

5. **Relu Function :**The rectified liner unit, or ReLU, is a non-linear activation function commonly employed in neural networks. It may be mathematically defuned as: **f(x) = max (0,x)**.

6. **Leaky Relu Function :** Leaky ReLU is an improvised version of the ReLU function in which, instead of declaring the ReLU function's value as zero for negative values of x, it is defined as an extremely tiny linear component of x. It may be expressed numerically as: **f(x) = 0.01x, x** $<$ **0** and **f(x) = x, x** $\geqslant$ **0**.

7. **Swish Function :**Researchers at GOOGLE found the Swish function, which is a relatively novel activation function. Swish outperforms the ReLU function in some circumstances. Mathematically, it is written as:**f(x) = x * sigmoid(x)**and **f(x) = x/(1** $-$ exp $(-x)$**)**.

8. **Softmax Function :**The Softmax function is a sigmoid function that has been combined with other sigmoid functions. Because a sigmoid function gives values in the range of 0 to 1, they might be interpreted as probability of data points in a specific class.

3.3.4 Loss Function :

3.3.4.1 What is an Loss Functions ?

The loss function is a function that calculates the difference between the algorithm's actual output and the predicted output. It's a way of determining how well your algorithm models the data. It may be divided into two categories. The first is for classification (discrete values, 0,1,2,...), while the second is for regression (continuous values) [8].

3.3.4.2 Types Of Loss Functions :

the commonly used loss functions to train a Neural Network are :

- Classification :

  1. **Cross-entropy :**
     - **Definition :** The purpose of this function is to quantify the difference between two averages of the amount of bits in a distribution of information, which originates from information theory. The difference between two probability distribution functions is computed using the cross-entropy as the Log Loss function (not the same, but they measure the same thing).
     - **Types of cross-entropy :**
       (1) Binary cross-entropy: for binary classification problem
       (2) Categorical cross-entropy: binary and multiclass problem, the label needs to be encoded as categorical, one-hot encoding representation (for 3 classes: [0, 1, 0], [1,0,0]…)
       (3) Sparse cross-entropy: binary and multiclass problem (the label is an integer — 0 or 1 or … n, depends on the number of labels)
     - **Range of values for this class of Loss function:** 0.00: Perfect probabilities
       $<$ 0.02: Great probabilities
       $<$ 0.05: In a good way
       $<$ 0.20: Great
       $>$ 0.30: Not great
       1.00: Hell
       $>$ 2.00 Something is not working

  2. **Log-Loss :**
     - **Definition :** The Binary cross-entropy up to a factor 1 / log is the Log-Loss. For negative values, this loss function is convex and increases linearly (less sensitive to outliers). The logistic regression is a typical approach that employs the Log-loss.

[8] https://towardsdatascience.com/what-is-loss-function-1e2605aeb904

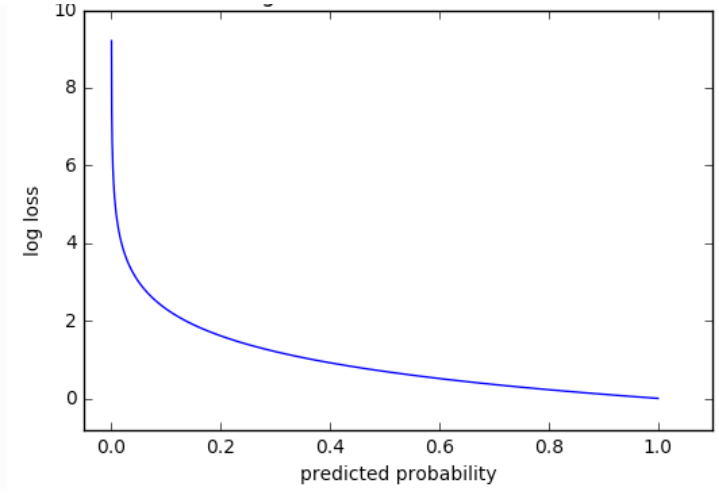

Figure 3.12: Log Loss function.

3. **Exponential Loss :**
   - **Definition :** The exponential loss was design at the beginning of the Adaboost algorithm which greedily optimized it. The mathematical form is: exp_loss = 1/m * sum(exp($-y * f(x)$))

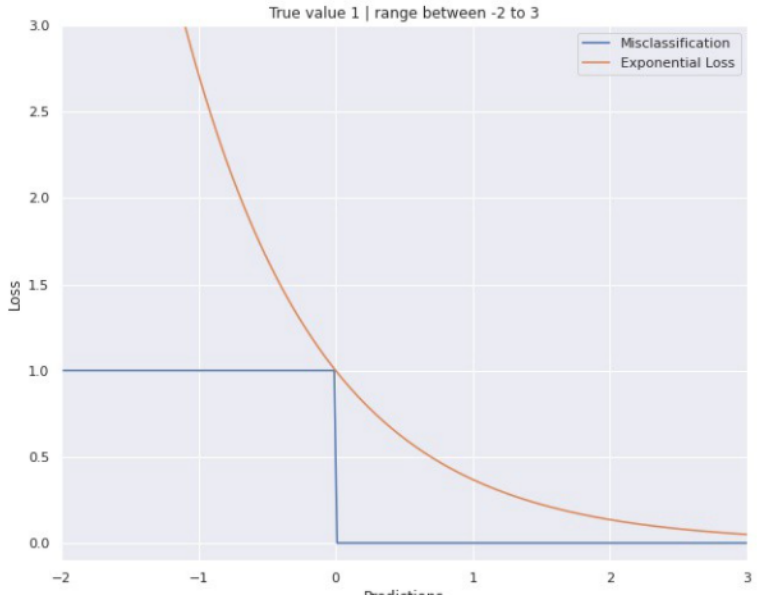

Figure 3.13: Exponential Loss .

4. **Hinge Loss :**
   - **Definition :** The Hinge loss function was created to fix the SVM algorithm's hyperplane in classification tasks. The purpose is to apply different penalties at points where the hyperplane is not precisely anticipated or is excessively closed.= max(0, 1-y*f(x)) is the mathematical formula for it.

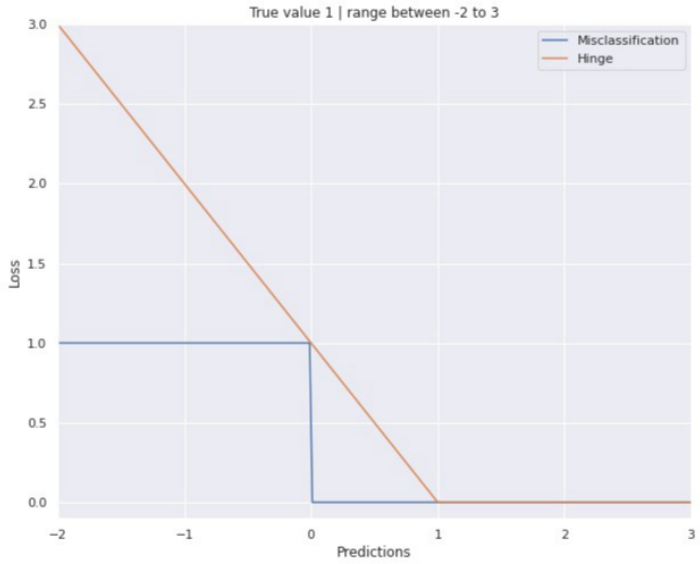

Figure 3.14: Hings Loss .

5. **Kullback Leibler Divergence Loss :**
   - **Definition :** The score of two distinct probability distribution functions is the KL divergence. It's worth noting the KL difference between a PDF of q(x) and a PDF of p(x). KL(Q||P), with || denoting divergence.(it is not symmetric KL(P||Q) != KL(Q||P)).KL(Q||P) = -sum(q(x) * log(p(x)/q(x)) or sum(q(x)*log(q(x)/p(x))

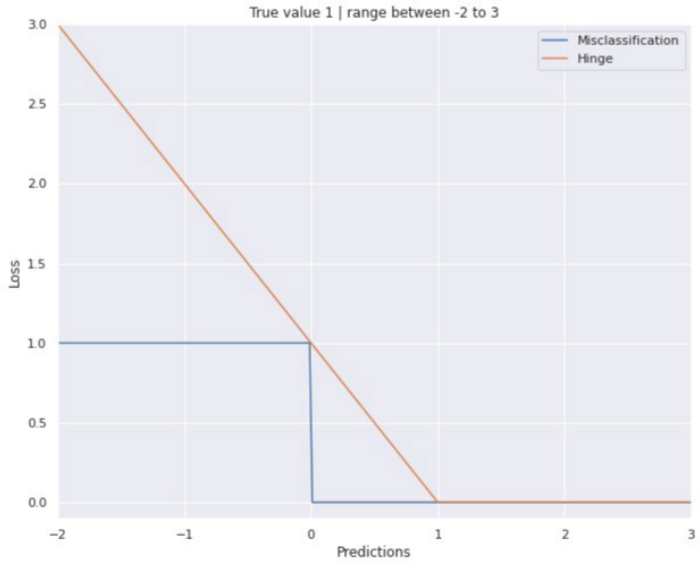

Figure 3.15: Exponential Loss .

- **Regression :**
  1. **Mean Square Error Loss**
     - **Definition :** It's the square of the difference between the actual y pred and the predicted y true outputs divided by the number of outputs. Outliers are

extremely sensitive to the MSE function since the difference is a square, which gives outliers greater weight. The mean should be the forecast if we were to estimate one value for all objectives.

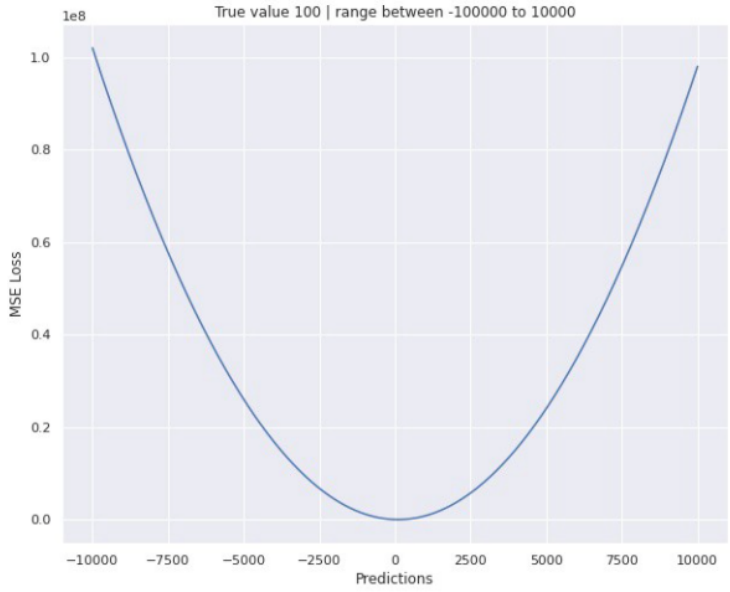

Figure 3.16: MSE Loss .

2. **Mean Absolute Error Loss**
   - **Definition :** The MAE function is more robust to outliers because it is based on absolute value compared to the square of the MSE. It's like a median, outliers can't really impact her behavior. The gradient is the same at each point, even when the values are closed to the minima.

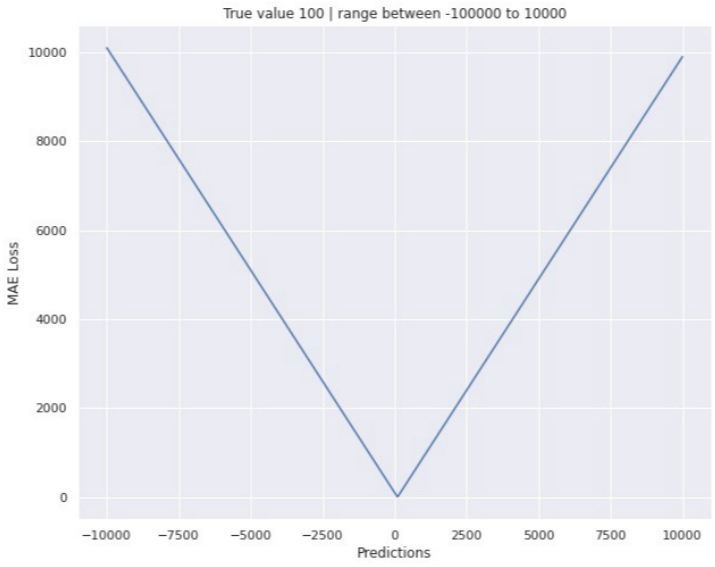

Figure 3.17: MAE Loss .

3. **Mean Square Logarithmic Error**

– **Definition :** The Huber Loss is a hybrid of MAE and MSE , with an extra parameter called delta influencing the form of the loss function. The algorithm will need to fine-tune this parameter. The function acts like the MAE when the values are high (far from the minima), and like the MSE when the values are small (near to the minima). Your sensitivity to outliers is represented by the delta parameter. The Huber Loss has the following mathematical form:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \leq \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

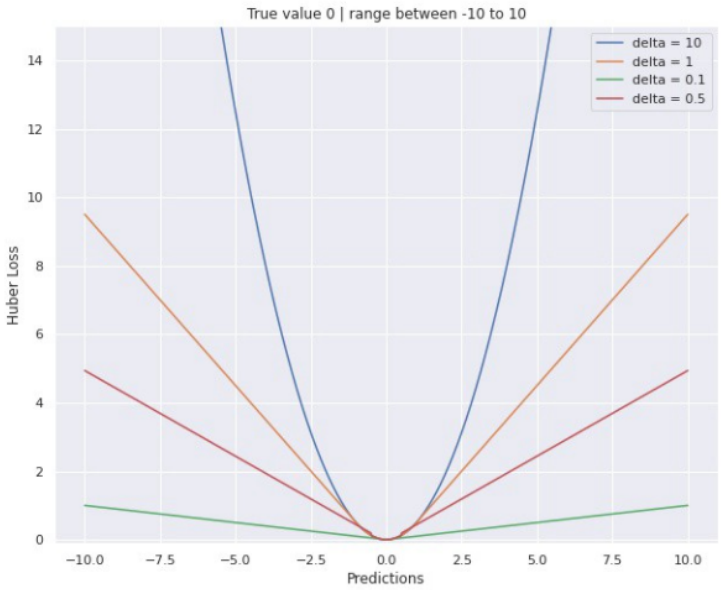

Figure 3.18: Huber Loss .

3.3.5 Optimizers in Deep Learning :

3.3.5.1 What is an optimizer?

Optimizers are techniques or strategies for minimizing an error function (loss function) or increasing production efficiency. Optimizers are mathematical functions that are based on the learnable parameters of a model, such as Weights and Biases. Optimizers assist in determining how to adjust the weights and learning rate of a neural network in order to minimise losses(Mustapha 2021).

3.3.5.2 Types of optimizers (Mustapha 2021) :

1. **Gradient Descent**

   - **Definition:** Gradient descent is a convex feature-primarily based totally optimization approach. It iteratively adjusts its settings on the way to lessen a given feature to its nearby minimum. Gradient Descent decreases a loss feature again and again with the aid of using visiting with inside the contrary route of sharpest ascent.

$$W_{new} = W_{old} - \alpha * \frac{\partial(Loss)}{\partial(W_{old})}$$

   - **Advantages of Gradient Descent :**
     - Easy to understand
     - Easy to implement
   - **Disadvantages of Gradient Descent**
     - the entire data set in one update, the calculation is very slow.
     - requires large memory and it is computationally expensive.

2. **Learning Rate :**

   - **Definition :**The learning rate, which determines how fast or slow we will advance towards the ideal weights, determines how big or tiny the steps are taken by gradient descent in the direction of the local minimum.

3. **Stochastic Gradient Descent:**

   - **Definition :**Gradient Descent is a version of this game. It does a one-by-one update of the model parameters. SGD will update the model parameters 10 times if the dataset is 10K.
   - **Advantages of Stochastic Gradient Descent:**
     - Frequent updates of model parameter
     - Requires less Memory.
     - Allows the use of large data sets as it has to update only one example at a time.
   - **Disadvantages of Stochastic Gradient Descent**

- The frequent can also result in noisy gradients which may cause the error to increase instead of decreasing it.
- High Variance.
- Frequent updates are computationally expensive.

4. **Mini-Batch Gradient Descent :**

- **Definition :** It is a hybrid of SGD and batch gradient descent techniques. It absolutely divides the education dataset into small batches and updates every batch separately. This moves a compromise among stochastic gradient descent's resilience and batch gradient descent's efficiency. When the parameters are adjusted, the variance is reduced, and the convergence is greater stable. It divides the statistics series into batches of fifty to 256 randomly decided on instances.
- **Advantages of Mini-Batch Gradient Descentt:**
  - It leads to more stable convergence.
  - Requires less Memory.
  - Requires less amount of memory.
- **Disadvantages of Mini-Batch Gradient Descent**
  - Mini-batch gradient descent does not guarantee good convergence,
  - If the gaining knowledge of fee is simply too small, the convergence fee might be slow. If it's far too large, the loss feature will oscillate or maybe deviate on the minimal value

3.3.5.3 How to choose optimizers ?

- Find a related research paper and start with using the same optimizer.
- Compare properties of your dataset to the strengths and weaknesses of the different optimizers.
- Adapt your choice to the available resources.

3.3.6 Evaluation Metrics

3.3.6.1 What are Evaluation Metrics?

Evaluation metrics are used to degree the high-satisfactory of the statistical or gadget mastering model. Evaluation of fashions or gadget mastering algorithms is important to any project. There are many styles of assessment metrics to be had to check a model (Mishra 2018).

3.3.6.2 Why is this Useful?

It is important to assess your version the usage of quite a few assessment indicators. This is because of the reality that a version may also carry out nicely whilst the usage of one size from one assessment meter however badly whilst the usage of some other size from a extraordinary assessment metric. In order to make sure that your version is jogging efficiently and optimally, you should use evaluation metrics [9]

3.3.6.3 Types of Evaluation Metrics (Agarwal 2019):

- **Confusion Matrix** A confusion matrix is an N X N matrix, wherein N is the quantity of instructions being predicted. For the trouble in hand, we've got N=2, and therefore we get a 2 X 2 matrix. Here are some definitions, you want to bear in mind for a confusion matrix :
  - True Positives (TP):Cases in which the forecast is correct and the actual value is correct.
  - True Negatives (TN) :the situations in which the forecast is negative and the actual value is negative.
  - False Positive (FP):When a positive forecast is made but the actual number is negative.
  - False Negative (FN) :when the forecast is negative but the value is positive.
- Accuracy :The most basic categorization metric is accuracy. It's really simple to comprehend. And it's well-suited to both binary and multiclass classification problems.
  Accuracy = (TP+TN)/(TP+FP+FN+TN)
- Precision :Let's start with precision, which provides a response to the following question: what percentage of anticipated Positives are actually Positive?
  Precision = (TP)/(TP+FP)
- Recall : Another relevant metric is recall, which addresses a different question: what percentage of real Positives is identified correctly?
  Recall = (TP)/(TP+FN)
- F1 Score:The F1 score is the harmonic mean of accuracy and recall, and it ranges from 0 to 1.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

[9] https://deepai.org/machine-learning-glossary-and-terms/evaluation-metrics

$$DCG_N = \sum_{i=1}^{N} \frac{rel_i}{\log_2(i+1)}$$

- Log Loss/Binary Crossentropye:For binary classifiers, log loss is an appropriate assessment measure, and it's also the optimization goal in cases like logistic regression and neural networks.
- Categorical Crossentropy:The log loss is also applicable to multiclass problems. In a multiclass situation, the classifier must give a probability to each class for each case.
- AUC:The area under the ROC curve is referred to as the AUC. The AUC ROC reveals how effectively the probabilities from the positive and negative groups are separated.
- Normalized discounted cumulative gain : To evaluate the anomaly detection algorithms, we propose using a metric called the normalized discounted cumulative gain metric (or nDCG for short). It is a metric often used in information retrieval to assess the quality of a ranking. Given a typical document search application, one could argued that, from a user's perspective, relevant documents are more valuable to a user than marginally relevant documents and a relevant document ranked high in the returned list of results is more valuable than an equally relevant document ranked lower in the list. A user may be reasonably assumed to scan the list of returned results from the beginning before interrupting the scan at some point correlated with time availability, effort required as well as the cumulated information from documents already seen. So it is safe to assume that relevant documents located further down the list of returned results are unlikely to be seen by the user as they would require more time and effort and become less valuable. Taking these facts into account, Järvelin and Kekäläinen introduced the nDCG measure.

  We similarly argue that, processes that are part of an attack but are ranked very low by an anomaly detection technique are virtually useless to an analyst since his/her monitoring burden would increase substantially with the amount of processes to be checked (not to talk about issues such as acquired loss of trust in the automated monitoring system and discarding of its alerts as well as the increased potential for misses and errors with the increase of data to monitor). Because of this, we believe nDCG to be an appropriate metric for our application

  To compute the nDCG, one could start by computing a score called discounted cumulative gain or DCG. The basis of DCG is that

  each document/entity in the ranking is assigned a relevance score and is penalized by a value logarithmically proportional to its position/rank in the list of results. The DCG is therefore computed as follows:

  where N is the number of entities/documents in the list, reli the relevance score of the ith entity/document in the list.

  Since the length of result lists can vary and the DCG score does not take that into account, it is common to normalize the DCG score by the ideal DCG score (iDCG), which is simply the best achievable DCG score, i.e. the score that would be achieved if all relevant entities were at the top of the list (and in the case of different degrees of relevance, with the highest values of relevance at the very top). Assuming we have p relevant entities in the list, we have:

$$iDCG_N = \sum_{i=1}^{N} \frac{rel_i}{\log_2(i+1)} \quad nDCG_N = \frac{DCG_N}{iDCG_N}$$

In our case, we only consider entities to be either relevant (processes that are part of an attack) or irrelevant (processes with normal behavior) and assign a relevance score reli of 1 to attack processes and of 0 to benign processes, and the idealized score results from ranking all k attack processes at positions 1, . . . , k.The closer the nDCG score to 1, the better the ranking.

CHAPTER 4

ANOMALY DETECTION

Anomaly detection is a critical subject that has been studied in a variety of fields and application domains. Many anomaly detection approaches have been tailored to specific application areas, while others are more universal.
The aim of this chapter is to definite the anomaly detection and take a Locke about its types and so on.

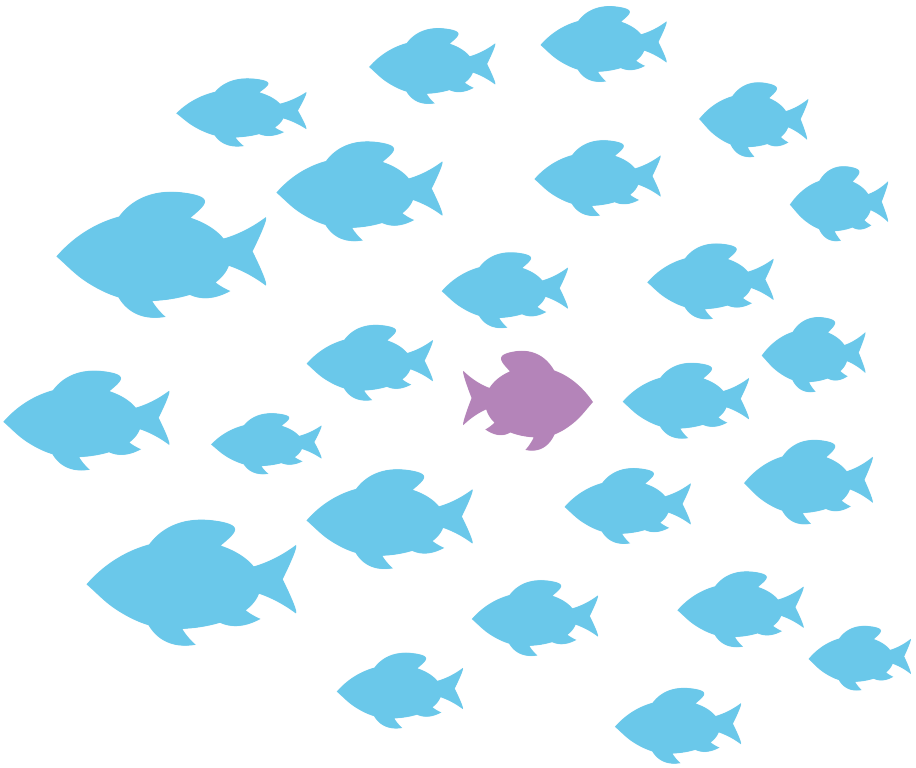

Figure 4.1: Anomaly detection

4.1 Anomaly Detection:

4.1.1 What Is Anomaly Detection?

The difficulty of discovering patterns in data that do not conform to anticipated behavior is known as anomaly detection. In various application fields, these non-conforming patterns are referred to as anomalies, outliers, discordant observations, exceptions, aberrations, surprises, oddities, or contaminants(Chandola, Banerjee, and Kumar 2009b).

4.1.2 Types of Anomaly Detection?

- **Point Anomaly:**When an individual data instance may be regarded aberrant in comparison to the rest of the data, it is referred to as a point anomaly. This is the most basic sort of anomaly, and it is the subject of the majority of anomaly detection research.

- **Contextual Anomaly:** Contextual anomalies, also known as conditional anomalies, occur when a data instance behaves abnormally in one context but not in another.

- **Collective Anomaly:** A collective anomaly occurs when a group of connected data examples is abnormal in comparison to the overall data set. Individual data occurrences in a collective anomaly may not be abnormal in and of itself, but their presence as a group is.
  (Goldberger et al. 2000) gives an example of a human ECG output in Figure 3.2. Because the same low value remains for an unusually long time, the highlighted region signifies an abnormality (corresponding to an Atrial Premature Contraction). It's worth noting that the low number isn't an exception in and of itself.

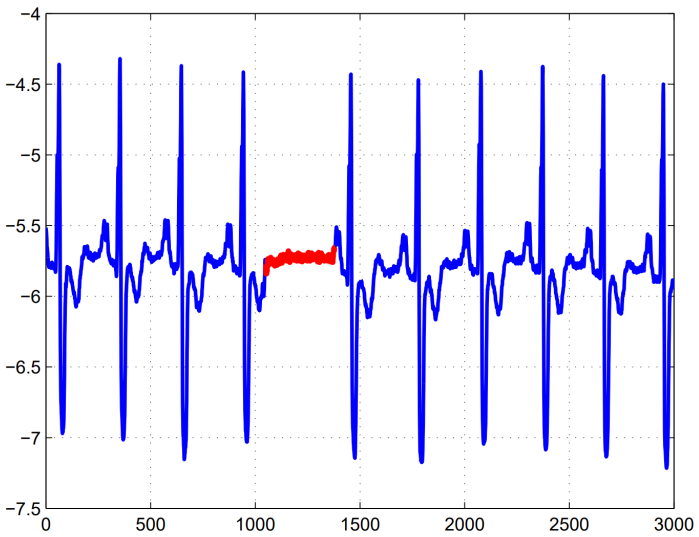

Figure 4.2: Collective anomaly corresponding to an Atrial Premature Contraction in an human electrocardiogram output (Goldberger et al. 2000)

4.1.2.1 Data Labels :

The labels attached to a data instance indicate whether it is normal or abnormal. Anomaly detection algorithms can function in one of three modes, depending on the extent to which labels are available (Chandola 0200):

- **Supervised anomaly detection :** In supervised mode, techniques presuppose the presence of a training data set with labeled cases for both normal and anomalous classes. In such instances, a typical method is to create a prediction model for normal vs. anomalous classes. Any data instance that hasn't been seen before is compared to the model to identify which class it belongs to. In supervised anomaly detection, there are two primary concerns to consider. First, the anomalous cases in the training data are significantly less than the regular instances.

- **Semi-Supervised anomaly detection :** Semi-supervised techniques presume that the training data only contains labeled cases for the normal class. They are more extensively applicable than supervised procedures since they do not require labels for the anomaly class. An accident would be represented by an anomaly scenario, which is difficult to model. Building a model for the class matching to normal behavior and using the model to spot anomalies in the test data is a common strategy employed in such strategies.

- **Unsupervised anomaly detection :**Unsupervised techniques don't require any training data and are thus the most extensively used. The strategies in this category make the implicit assumption that typical cases in the test data are significantly more common than abnormalities. If this assumption is incorrect, such systems have a high probability of false alarms. By employing a sample of the unlabeled data set as training data, many semi-supervised approaches may be converted to function in an unsupervised mode. Such adaptation is based on the assumption that the test data contains few abnormalities and that the model learned during training is resistant to these few anomalies.

4.1.2.2 Output of Anomaly Detection :

The way in which abnormalities are reported is a key feature of any anomaly detection technique. Anomaly detection techniques often provide one of the following two sorts of outputs (Chandola 0200):

- **Scores :**Abnormality scores are assigned to each occurrence in the test data based on the degree to which that instance is deemed an anomaly by scoring procedures. As a result, such approaches produce a ranked list of anomalies. An analyst can study the top few abnormalities or choose the anomalies using a cut-off criteria.

- **Labels :**This group of techniques assigns a label (normal or abnormal) to each test occurrence. Internally, several ways generate a score for each test instance and give a label using either a threshold or a statistical test. An analyst can utilize a domain-specific threshold to choose the most relevant abnormalities using scoring-based anomaly detection algorithms. Techniques that assign binary labels to test cases do not directly allow analysts to make this decision, although it can be influenced indirectly by parameter selection within each technique.

4.1.2.3 Anomaly Detection challenges :

Anomaly detection techniques, as we have seen, are based on the concept of modeling what is normal and expected, and then try to detect what is not. Although this approach seems simple, it faces real difficulties and challenges, for example:

1. The boundary between normal and abnormal data is not always precisely defined. For example , in the Figure 4.3, if the point O2 is more close to the boundary of the region N2 , it can be a normal or anomalous. Thus, the selection and adjustment of a threshold may require some domain expertise.

2. The lack of data sets for training and validating anomaly detection models.

3. The nature of the data may change over time, and what is normal in the present may not be in the future.

4. In the field of cybersecurity, attacks like credit card fraud, spamming and phishing attacks tend to appear like normal actions. Detecting anomalous data in this situation is difficult and need complex and adaptive models.

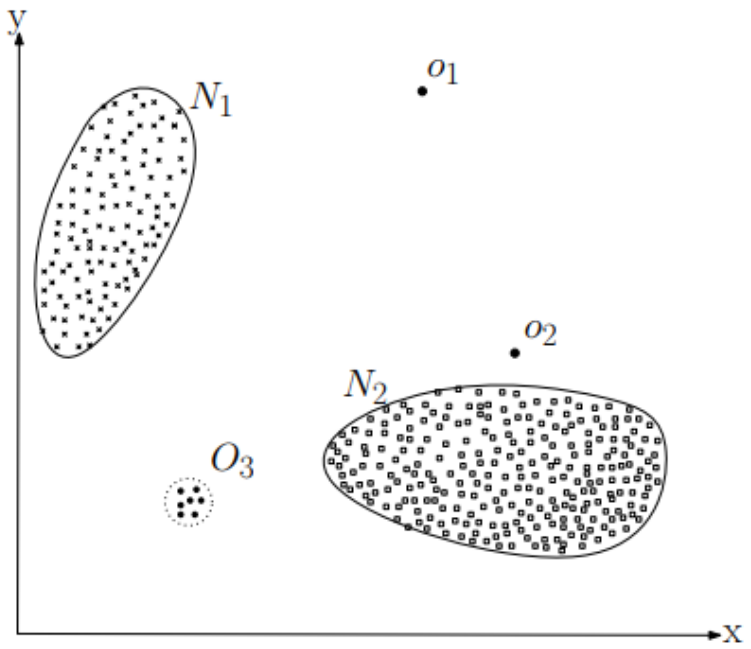

Figure 4.3: Illustration of anomalies in two-dimensional data set (Chandola, Banerjee, and Kumar 2009a)

4.2 Conclusion :

Anomaly detection is critical and beneficial in a variety of real-world applications, such as detecting computer network breaches and detecting fraud. In this chapter we gave a very simple definition of Anomaly detection their types ,types of their output and so on and analyzed the research efforts to understand how to tackle and address the challenges in the field of context-aware computing.

Part III

State of Art

CHAPTER 5

ANOMALY DETECTION APPROACHES

In this section, we will present a summary of some of these approaches. They are classified according to the learning method being used, whether it is Unsupervised ,Supervised and Semi-supervised. At the end, we present a summary table of the reviewed approaches.

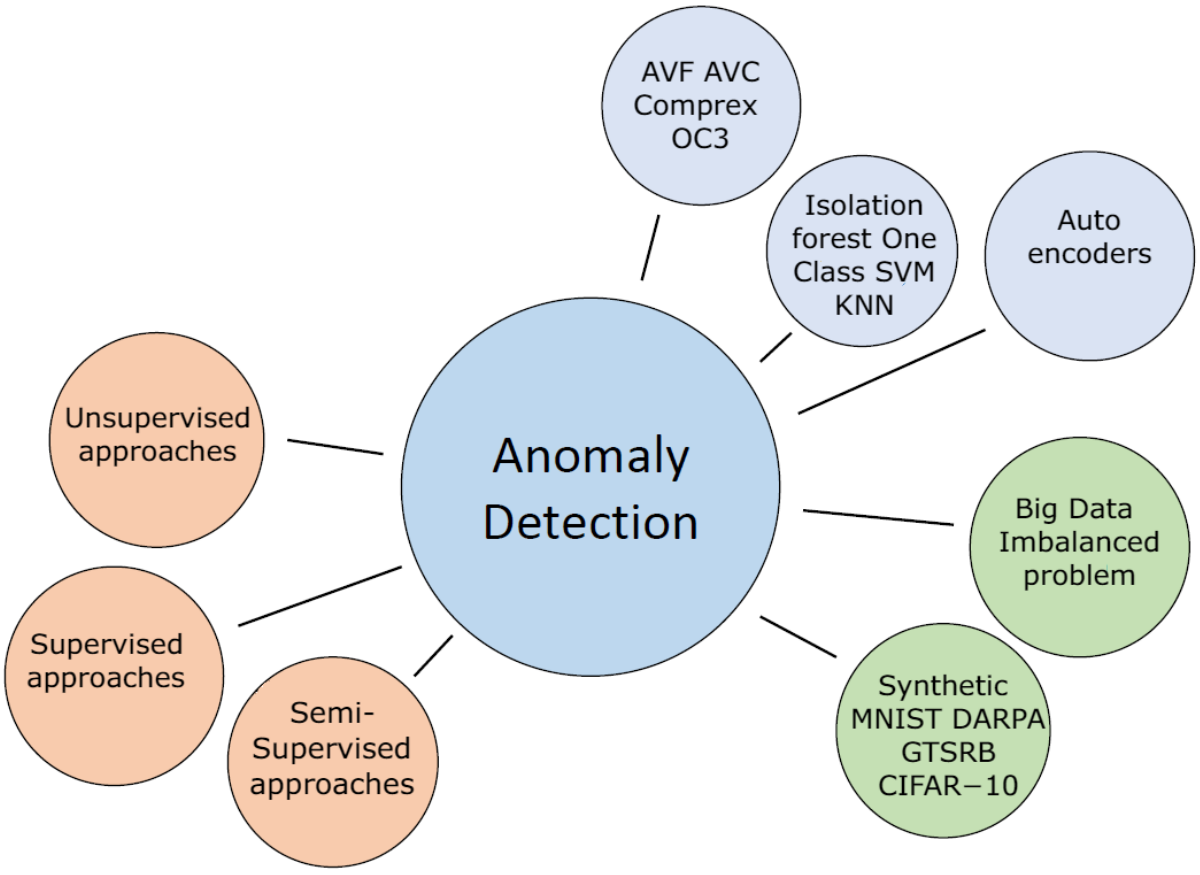

Figure 5.1: Anomaly Detection Methods

5.1 Benchmark intrusion dataset :

Due to privacy issues, datasets used for network traffic analysis in commercial products are not easily available. However, there are a few public widely used datasets that play a vital role in the evaluation and validation of any IDS approach.

5.1.0.1 DARPA / KDD 98

In 1998, DARPA (Defence Advanced Research Project Agency) created the KDD98 (Knowledge Discovery and Data Mining (KDD)) dataset [Hem20]. Although this dataset was an important contribution to the research on IDS, its accuracy and capability to consider real-life conditions have been widely criticized. This dataset was collected using multiple computers connected to the Internet to model a small US Air Force base of restricted personnel. The collected network packets were around four gigabytes containing about 4,900,000 records. The test data had around 2 million connection records, each of which had 41 features plus an additional one for the label (Khraisat et al. 2019).

5.1.0.2 KDD Cup99 :

The 1998 DARPA Dataset was used to create the KDD Cup99 dataset , which was utilized in the Third International Knowledge Discovery and Data Mining Tools Competition.

5.1.0.3 NSL-KDD :

NSL-KDD is a publicly available dataset that evolved from the KDD cup99 dataset. A statistical examination of the KDD cup99 dataset revealed significant flaws that have a significant impact on intrusion detection accuracy and lead to a skewed assessment of IDS (Tavallaee et al. 2009). The biggest issue with the KDD data set is the large number of duplicate packets, with around 78 percent of network packets in both the training and testing sets being duplicated. There are 125,973 records in the NSL-KDD train set and 22,544 records in the test set. The size of the NSL-KDD dataset is large enough that it is feasible to use the entire dataset without sampling at random. The NSL KDD dataset contains 42 characteristics and 22 training intrusion assaults (i.e., features).

5.2 Comparison between the datasets :

		DARPA	**KDD CUP 99**	**NSL-KDD**
General Information	Year of Creation	1998/99	1998	1998
	Public Availability	yes	yes	yes
	Normal Traffic	yes	yes	yes
	Attack Traffic	yes	yes	yes
Natrure Of Data	Frormat	packet , logs	other	other
	Anonymity	none	none	none
Data Volume	Count	5M points	5M points	150K points
	Duration	7/ weeks	n.s	n.s
DRecording Environment	Kind of Traffic	emulated	emulated	emulated
	Type of Network	small network	small network	small network

Table 5.1: Comparison between the datasets

5.3 Unsupervised approaches :

James Cheney in Cheney et al. 2020 proposed a MDL-based anomaly detection model that can deal with heterogeneous data,By fitting a mixed model to the data using a form of k-means clustering. They proceeded as follows:

- Using the MDL principle, create a decent model of the data.
- Using the model, assign an anomalous score to each element based on its compressed size.
- Examine the data items with the highest scores to see which ones are the most unusual.
- Finally, we present a general technique (i.e., a meta-algorithm) for learning MDL hypotheses mixtures.

For that they implemented AVC and AVC∗ in Python, using libraries for linear algebra to perform the iterative clustering steps efficiently. They also implemented Python scripts that run adapted versions of Krimp/OC3 and CompreX to perform their clustering variants OC3∗ and CompreX∗ .

For that, they consider several public datasets collected for evaluation of categorical anomaly detection by Pang that contained 2k records and between 22–114 attributes, with between 27 and 60 anomalies after that they transformed all datasets to use binary encodings of multivalued attributes to ensure compatibility with Krimp,then they choosing one class to be the normal class and selecting a few examples of another class to be the anomalies ,they also consider several datasets derived from the DARPA Transparent Computing program, consisting of data about operating system processes in a system under attack by an advanced persistent threat (APT).

For the comparison they used Both nDCG and AUC scores because AUC score is not very informative because they have large datasets with very sparse anomalies.

Several experiments that have been done to test the performance of the model are presented in the next table :

				Kromp		CompreX		AVC		AVC*	
Dataset	N	M	% Anomaly	AUC	nDCG	AUC	nDCG	AUC	nDCG	AUC	nDCG
AID62	4.279	114	1.4%	0.581	0.409	**0.675**	0.423	0.644	0.420	0.674	**0.433**
Bank	41.188	52	11%	0.625	0.814	**0.639**	**0.823**	0.593	0.808	0.608	0.810
Chess(KRK)	28.056	40	0.01%	0.321	0.220	0.622	**0.263**	0.0.645	0.244	**0.673**	0.254
CMC	1.473	22	2.7%	0.559	0.402	0.580	**0.458**	0.589	0.474	**0. 600**	0.414
Probe	64.759	82	6.4%	0.938	0.925	0.937	0.915	**0.951**	**0.961**	0.937	0.912
SolarFlare	1.066	41	4%	0.792	**0.593**	**0.837**	0.588	0.826	**0.593**	0.783	0.545
Windows S1	17.569	22	0.04%	0.992	0.302	**0.996**	0.602	0.984	0.618	**0.996**	**0.675**
BSD S1	76.903	29	0.02%	**0.976**	0.436	**0.976**	**0.542**	0.882	0.525	0.975	0.516
Lunix S1	247.160	24	0.01%	**0.887**	0.340	**0.887**	0.299	0.821	0.264	**0.887**	**0.407**
Android S1	102	21	8.8%	0.754	0.740	0.731	0.821	0.826	0.848	**0.860**	**0.861**
Windows S2	11.151	30	0.07%	0.857	**0.242**	0.856	0.223	0.808	0.230	**0.881**	**0.240**
BSD S2	224.624	31	0.004%	**0.936**	0.436	**0.249**	0.904	0.211	0.873	0.191	0.186
Lunix S2	282 .877	25	0.01%	0.873	0.387	**0.873**	**0.469**	0.824	0.306	0.856	0.358
Android S2	12.106	27	0.1%	0.884	0.328	**0.930**	**0.780**	0.906	0.305	0.907	0.629

Table 5.2: The performance of the model

As a summary, in this study, they found that because AVC is algorithmically a lot simpler than Krimp and has numerous untapped parallel processing capabilities, future optimization or parallelization should be successful.

In this research Benabderrahmane et al. 2021, the authors presented the results of an unsupervised method that exploits OS-independent features reflecting process activity to detect realistic APT-like attacks from provenance traces.

For this purpose they designed two separate anomaly detection methods: VRARM for Valid Rare ARM and VF-ARM for Valid Frequent ARM. The underlying association rules have been derived from MRIs and MFIs, respectively.

They employed two publicly available data sets mentioned in (Berrada et al. 2020) for their evaluation. These "scenarios" are made up of sets of feature views or contexts extracted from raw whole-system provenance graphs created during two DARPA Transparent Computing (TC) program "engagements".

They evaluated the classifier using the AUC rate, nDCG rate, and the results are shown in the following table:

		Winner AD method		**Winner context**		**Running time (sec)**		**nDGG**		**AUC**	
	Attack scenario	1	2	1	2	1	2	1	2	1	2
Source	Windows	VR-ARM	VR-ARM	ProcessEvent	ProcessAll	4.60	23.8	0.82	0.35	0.75	0.50
	BSD	VR-ARM	VR-ARM	ProcessEvent	ProcessNetflow	12.18	12.30	0.64	0.60	0.75	0.50
	Linux	VR-ARM	VR-ARM	ProcessNetflow	ProcessAll	3.53	2.74	0.58	0.45	0.83	0.50
	Android	VR-ARM	VR-ARM	ProcessEvent	ProcessNetflow	0.78	3.35	0.87	0.71	0.92	0.50

Table 5.3: Classifier using the AUC ratE, nDCG rate

Berrada et al. 2020 proposed an approach that summarizes process activity using categorical or binary features such as the kinds of events performed by a process.The main contributions of this paper are:

- For the objective of identifying APT-like behavior in system provenance traces, baseline findings were established for five categorical anomaly detection algorithms, namely FPOF, OD, OC3, CompreX, and AVF (in both batch and streaming modes for AVF).
- Examining and comparing the efficacy of these five anomaly detection methods for the task at hand.
- Showing that some methods, such as OC3 and AVF, can already produce useful detection results in reasonable time despite their relative simplicity ("naive" set of features requiring little domain knowledge or tweaking and/or very simple anomaly scoring strategy e.g. AVF) and that these results can be easily replicated in a streaming setting in some cases (e.g. AVF).
- Discussing various metrics for the detection problem and suggesting that a measure from information retrieval (normalized discounted gain) be used

For this evaluation they used two data collections representing two attack scenarios, each consisting of several days' worth of activity in a DARPA evaluation of provenance-tracking systems, running on Windows, BSD, Linux and Android respectively. They evaluated all of the above algorithms in batch mode . As a result the AVF has anomaly detection performance comparable or better than the itemset mining-based techniques, typically finding at least some parts of the attack within the top 1% or even 0.1%. They also compare batch and streaming AVF with different block sizes and the results show that there is little degradation in anomaly detection performance.

Akoglu in (Akoglu et al. 2012) they introduced a new approach for identifying anomalies using pattern-based compression(COMPREX).

For that work they used complex databases, including graph, image and relational databases that may contain both categorical and numerical features from diverse domains.
They assessed their system based on four factors: (1) bit compression cost, (2) running time, (3) detection accuracy, and (4) scalability.in order to find the optimal set of dictionaries that yield the minimal lossless compression, and then spot tuples with long encoded lengths.

Experiments show that their method COMPREX can do better, yielding both better compression (and relatedly, higher detection accuracy) as well as lower running time.Where it achieves very nice compression rates, outperforming KRIMP for all of the datasets, and providing up to 96% savings in bits (47% on average).

As a conclusion of this article they found that COMPREX is widely applicable; after discretization, it matches the state of the art for numerical data and properly detects abnormalities in huge graphs and picture data.

In the following work (Syarif, Prugel-Bennett, and Wills 2012), authors implemented and compared the performance of five different clustering algorithms in their anomaly detection module.
In the feature extraction part, they selected only numerical data and handled missing value, then they used the normalization to convert all attributes/variables to a common scale with an average of zero and a standard deviation of one.
The NSL-KDD intrusion dataset was used to train the system along with the following metrics for evaluation: accuracy rate and false-positive rate. The following table presents the evaluation results

Algorithm	Accuracy	Flase positive
K-means	57.81%	22.95%
Improved K-means	65.40%	21.52%
K-mesoids	76.71%	21.83%
EM clustering	78.06%	20.74%
Distance-based outlier detection	80.15%	21.14%

Table 5.4: Obtained Result using Accuracy and FP

The clustering algorithms are able to detect intrusions without prior knowledge. In this experiment, the distance-based outlier detection algorithm achieves the best accuracy with 80.15%, followed by EM clustering 78.06%, k-Medoids with 76.71%, improved k-Means 65.40% and k-Means 57.81%. This work shows that the clustering approach can reveal acceptable results in anomaly detection where we have seen that the distance-based outlier detection method outperforms the other four clustering algorithms. However, all of these algorithms didn't surpass the 80% accuracy rate.
In addition, they have a quite high false positive rate (more than 20%) making this anomaly detection module not suitable for real-world tasks. Therefore, future work is needed to improve the accuracy rate and reduce the false-positive rate.

The authors created a hybrid model that included both GA and SVM for this study (Aslahi-Shahri et al. 2016). When the GA method is used to reduce the amount of charac-

teristics and the SVM algorithm is used to determine if the scenario is normal or not. The GA-SVM feature selection algorithm is a mix of SVM and AG. To begin, GA creates feature subsets based on the original feature set. New feature subsets are then formed through a competitive mechanism of crossover and mutations, and only those with excellent fitness values (in this case, detection rates) are passed on to the next generation. This procedure is done until the best feature subset is found. The GA technique's phases are included in the learning moduleblock.
Finally, the best detection model is constructed using a collection of features and kernel function parameters. The model is eventually used to the categorization of new samples.
For evaluating the reliability of the suggested approach, a number of experiments were conducted using the KDD Cup99 dataset . The system has shown the best results. As presented in Table :

	True positive	Flase positive	Precision
GA-SVM	0.931	0.028	0.941

Table 5.5: Obtained result using TP ,FP ,Precision

As can be seen, the system produces an excellent outcome despite the fact that it only has four characteristics. The primary benefit of this research is the combining of two methodologies, which resulted in an optimal subset of characteristics with just four utilizing GA. However, because just four attributes may be completely indicative of network traffic, there is a risk of information loss. Furthermore, today's traffic differs significantly from that simulated in the old dataset "KDD CUP 99," therefore this system may struggle to defend against today's sophisticated assaults.

5.4 Supervised approaches :

Chawla, et al., 2019 Chalapathy, Menon, and Chawla 2018 proposed a propose a one-class neural network (OC-NN) model to detect anomalies in complex data sets .We summarized their main contributions as follows:

- For anomaly detection, they develop a new one-class neural network (OC-NN) model.
- For learning the parameters of the OC-NN model, they suggest an alternating minimization approach and they noticed that the OC-NN objective's subproblem is the same as solving a quantile selection problem.
- They conduct extensive tests that show that OC-NN outperforms other state-of-the-art deep learning algorithms for anomaly detection on complicated image and sequence data sets.

Additionally, they used four real-world datasets (Synthetic , MNIST , GTSRB ,CIFAR−10) and for each dataset, they performed a further processing to create a well-posed anomaly detection task.

The training of the neural network is driven by a one-class SVM-like loss function in OC-NN.

As a summery they show that OC-NN can achieve comparable or better performance in some scenarios than existing shallow state-of-the art methods for complex datasets, while having reasonable training and testing time compared to the existing methods.

In this research (Chu, Lin, and Chang 2019) proposes an attack detection method that allows for early identification of an APT assault.

For attack detection and verification, the system consults the NSL-KDD database.

They used The primary approach employs principal component analysis (PCA) to improve detection efficiency and feature sampling.

In this study, a comparison has been made between the correct rate of APT network attack detection using the NSL-KDD data sets and PCA dimensionality reduction technology and four machine learning classification algorithms: SVM, naive Bayes, decision tree,and the multi-layer perceptron neural network (MLP).

Finally, the pre-processed training and test data sets were grouped and tested,and experiments with the four classification algorithms were carried out.

As a rsult and according to the experiments in this study ,The support vector machine (SVM) had the greatest recognition rate, reaching 97.22 percent.

By converting traffic data to images, Wang et al. 2017 proposed malware traffic classification using a Convolutional Neural Network (CNN). This method, which was chosen and proposed, does not require any hand-designed features; instead, it takes raw data as an input and classifies it by transforming it to images. According to the author, this was the first attempt to categorize malware traffic using representations of raw data.

For testing purposes the authors created USTC-TFC2016 dataset and developed data-preprocessing toolkit USTC-TK2016. They mentioned that KDD-Cup99 and NSL-KDD offers multiple features but did not meet the require for raw data detection.

Comparison test results showed that Session + All representation had highest accuracy and average accuracy was respectful 99.41% .

The raw data approach could be used for APT detection without the traffic clear process.

Andropov et al. 2017 proposed multilayer perceptron Artificial Neural Network (ANN) with backspace propagation algorithm training in their paper.

For testing purposes the authors created USTC-TFC2016 dataset and developed data-preprocessing toolkit USTC-TK2016. They mentioned that KDD-Cup99 and NSL-KDD offers multiple features but did not meet the require for raw data detection.

The proposed method has two functional states: (i) offline and (ii) online traffic analysis and it uses data aggregation to detect patterns in the elseways highly variable traffic data.

For testing the proposed methods, a local ISP collected data for the authors from several hundred L2 nodes for a period of one month and this dataset was considered as a normal traffic.

The proposed ANN was able to detect both, known and unknown anomalies with high accuracy, however test results show that idle scan is the most difficult to detect. Custom anomaly had the lowest accuracy with 150000 classification iterations, ARP spoofing the second lowest and idle scan third lowest. After 300000 classification iterations all other anomalies had over 80% accuracy,while idle scan stayed below 80% [].

The proposed multi-output layer method gives more detailed information about the anomaly than the boolean alternative due to classification. This type of method can be useful in detecting APT attacks since it has multiple simultaneous attack vectors. It would be interesting to see how the idle scan detection accuracy could be increased, for example by optimizing the ANN, perhaps choosing different amount of hidden layers and neurons in a layer.

Authors of the following work (Vargas-Muñoz et al. 2018) used network flow records as an input to their system in order to detect anomalies. The open-source network audit tool Argus was used in this work to collect and analyze the three different type of network flows:

- Uniflow (Unidirectional Flow), which is composed only of packets sent from a single endpoint to another single endpoint.
- Biflows (Bidirectional Flow), which is composed of packets sent in both directions between two endpoints.
- Aggregated-flows, containing the flows going in both directions of an entire session.

Many traffic features were firstly observed and analyzed, then, a heuristical selection is performed to keep only the set of features that best represent the traffic behavior. Here are the eight flow-features that were selected for the creation of the Bayesian classifier: source IP,destination IP, protocol, source port, destination port, total packets, source bytes, destination bytes.
In the training phase, the Bayesian network was trained by a total of 763086 flows containing normal and anomalous traffic from a newly created dataset that represented as faithfully as possible the real traffic along with proper attacks. Also, the UNB ISCX IDS 2012 dataset, and the UAN W32.Worms 2008 dataset were used as benchmarks for evaluation purposes.
For evaluation purposes, the following metrics were used: True negative rate (TNR), True positive rate (TPR), False positive rate (FPR), False negative rate (FNR), and Accuracy. The evaluation of the Bayesian classifier results is shown in the following tables:

From the results obtained we can notice that the accuracy is high in all three cases, and the best result is achieved in the case of aggregate-flow. We can see that the highest accuracy was found in the case of uni-flows, however, aggregateflow has the highest TNR and lowest FPR. The key advantage of this work is the adoption of a flow traffic level to avoid the issue of high data dimensionality along with making a considerable reduction in the studied

Flows	Rate				
	Accurancy	TNR	TPR	FPR	FNR
Unflows	97.24	99.96	86.64	0.03	13.35
Biflows	99.83	100	98.71	0	0.74
AggregatedFlows	99.98	100	99.96	0	0.03

Table 5.6: Evaluation Results On Unb Iscx Ids 2012 Dataset

Flows	Rate				
	Accurancy	TNR	TPR	FPR	FNR
Unflows	99.95	99.96	99.31	0.03	0.68
Biflows	99.92	99.94	98.71	0.05	1.28
AggregatedFlows	99.92	99.99	98.35	0.007	1.64

Table 5.7: Evaluation Results On Uan W32.Worms 2008 Dataset

features which resulted in much less computational complexity. However as a disadvantage, there is a considerable loss of information when working at the flow level, since it compacts so much certain information. Another thing to notice is the lack of information about the dataset used during the training phase. if this dataset was originally inspired by the UNB ISCX IDS 2012 and the UAN W32.Worms 2008, performing the evaluation on these same two datasets shows really high-performance results (Accuracy > 99) which might be the case of system overfitting

Shen and Chow 2020 picked a neural network with 6 inputs, 10 neurons in the hidden layer, and 7 output neurons, each indicating an abnormality the system can identify, with the sixth neuron representing a ”unknown” anomaly and the seventh neuron representing ”regular” behavior. A sigmoid activation function is used by both output and buried layer neurons. They also utilized the backpropagation algorithm (Demuth et al. 2014) to train the network. The two researchers employed the Netflow protocol and certain aggregation criteria to solve the problem of massive network traffic. The following characteristics were considered in the study:

- Source and destination addresses
- Source and destination ports
- Protocol type
- Number of packets
- Size of packets

For this project, data was collected for one month from the local ISP network. Before any changes are made, the dataset acquired this way is regarded to demonstrate ”normal” behavior. Several small-scale abnormalities were constructed and injected into this dataset for

testing purposes: DDoS UDP flood - DDoS TCP flood - Port scan - Idle scan DDoS UDP flood - DDoS TCP flood - Port scan - Idle scan - ARP spoofing and the ”custom” anomaly, which was designed to see how well a neural network could detect an unknown class of abnormalities.
Table shows the results for 300000 iterations for various anomalies. The neural network,

Type	Accuracy	Quantity in dataset
DDos UDP flood	0.91	10
DDos TCP flood	0.81	10
Port scan	0.96	10
idle scan	0.78	8
ARP spofing	0.83	8
custom	0.85	5
normal	0.96	-

Table 5.8: Classification result after 300000 iterations

according to the chart, shows promise in recognizing both known and new forms of abnormalities. The implementation of ANN allows the system to quickly adapt to incremental network changes, which is the major benefit of this study. Another benefit of adopting the NetFlow protocol is that it overcomes the challenges of acquiring appropriate training data and filters network noise. However, by collecting a better-represented dataset and improving the aggregation criteria, this work may be improved, resulting in higher accuracy and lower false-positive rates.

Calssifier	Classification accuracy and performance metrics				
	Total accuracy	Precision	Recall	F-score	ROC area
Water_tower_dataset					
GRYPHON	98.03%	0.980	0.980	0.980	0.980
SOCCADF	98.08%	0.981	0.981	0.981	0.994
OCC-SVM	98.01%	0.980	0.980	0.980	0.995
OCC-CD/CPE	96.75%	0.975	0.975	0.975	0.980
gas_dataset					
GRYPHON	97.68%	0.975	0.975	0.970	0.980
SOCCADF	98.82%	0.988	0.988	0.988	0.995
OCC-SVM	97.98%	0.980	0.980	0.980	0.990
OCC-CD/CPE	95.67%	0.960	0.960	0.960	0.975
electric_dataset					
GRYPHON	96.92%	0.970	0.969	0.969	0.985
SOCCADF	98.30%	0.983	0.983	0.983	0.999
OCC-SVM	97.63%	0.978	0.978	0.978	0.990
OCC-CD/CPE	97.02%	0.970	0.970	0.970	0.985

Table 5.9: Classification accurancy and perfomance metries

5.5 Semi Supervised approaches :

In this research (Demertzis, Iliadis, and Bougoudis 2020), the authors presented an semi-supervised anomaly detection system based on one class evolving spiking neural network.

They presented the Gryphon advanced intelligence system.

Gryphon is a Semi-Supervised Unary Anomaly Detection System for big industrial data which is employing an evolving Spiking Neural Network (eSNN) One-Class Classifier (eSNN-OCC).

The output model can detect very fast and efficiently, divergent behaviors and abnormalities associated with cyberattacks which are known as Advanced Persistent Threat (APT).

The data used in the training process(The water tower dataset, The gas dataset includes, the electric dataset) is connected to the normal operation of vital infrastructure.

the next table represents a Comparison between Gryphon and others algorithms:

Barceló-Rico, Esparcia-Alcázar, and Villalón-Huerta 2016 they aimed at developing classifiers that can help human experts to find APTs.The method presented in their wotk utilises the premise that APT-infected HTTP traffic will tend to be anomalous to help create a classifier for suspicious/non-suspicious behaviour.

The classifier will be trained on data that has been labeled by humans and then evaluated on a new batch of data to see how well it performs.

The methods used to train the classifier are Genetic Programming (GP), two Decision Tree Classifiers (DTC), namely CART and Random Forests, and Support Vector Machines (SVM).

Their method, thus, follows these steps:

- Choose the most unusual cases from the data (HTTP requests registered by a proxy) to form the dark set.
- Request that a human expert categorize these events as suspicious or non-suspicious.
- utilizing a portion of the labelled data from the dark set to train a classifier.
- Using the remaining data from the dark set, test the classifier you created in the previous step.

The next tables shows the classification results with labelled and unlabelled data for different methods:

- Lablled Data :

		GP(%)	CART(%)	RF(%)	SVM(%)
Training	Tp	85.86	93.94	97.98	
	FN	14	6.06	2.02	11.11
	TN	75.56	95.56	96.3	98.52
	FP	2.61	0.02	1.78	0
Testing	S	95.27	23.8	87.19	43.65
	NS	4.73	76.15	12.81	56.35

Table 5.10: Lablled Data

- UnLablled Data :

As a summery in In labelled case all methods had similar results. However, in the labelled case it was shown that SVM and CART method outperformed the other two. The authors used a Conventional Autoencoder-based system with 8 neurons in the hidden layer for this study Chen et al. 2018. Because the system was trained on the NSL-KDD dataset and the data was normalized, all of the input features were translated to the range [0, 1]. They also divided the whole NSL-KDD Dataset into network traffic types (TCP, UDP, and ICMP) and performed a 10-fold cross-validation for each. During the training phase, the average reconstruction error is optimized and thresholded to create the usual profile of valid

		GP(%)	CART(%)	RF(%)	SVM(%)
Training	Tp	81.82	90.91	87.87	89.39
	FN	18.18	9.09	12.12	10.61
	TN	97.39	99.98	98.22	100
	FP	2.61	0.02	1.78	0
Testing	Tp	84.85	81.82	54.55	75.76
	FN	15.15	18.18	45.45	24.24
	TN	97.36	99.87	97.65	99.86
	FP	2.64	0.13	2.35	0.14

Table 5.11: UnLablled Data

	Accuracy(%)	False-positive rate(%)
Autoencode	95.85	4.09

Table 5.12: Autoencode results

traffic data.
Table 3.9 shows the performance results of the system:

We can plainly see that the system functions admirably, with extremely high accuracy and a low false-positive rate that is acceptable.
The key benefit of this study is that it uses Autoencoders to produce a strong dimensionality reduction-based network anomaly detection that can readily capture non-linear correlations between features.
Jasmeen K. Chahal's suggested approach (Chahal and Kaur 2016) entails preparing NSL-KDD dataset instances in order to transform character data to numeric data. The K-mean approach was then used to cluster this dataset. Finally, the adaptive SVM is used to classify the clusters that have been created.
The following measures are utilized for evaluation: Accuracy, False Negative Rate (FNR), and False Positive Rate (FPR) (FPR). The system's experimental findings are shown in the next table :

	Accuracy(%)	FPR(%)	FNR(%)
K-mean-ASVM	98.47	0.53	76.63

Table 5.13: K-mean-ASVM results

When it comes to accuracy, false-positive rate, and false-negative rate, the semi supervised system definitely outperforms the competition. As a result, this semi supervised method has proven to be effective in boosting intrusion detection system performance.

5.6 Synthesis table: :

The rising worry about security in information and communication technology has made the anomaly detection domain a vast study subject, with many distinct techniques and approaches for this goal emerging throughout time, as discussed in previous sections. In this part, we compare and contrast the various APTs anomaly detection algorithms that have been developed. As a result, in order to make the whole comparison process easier to comprehend,The table below is a summary that compares the different approaches cited in the previous section. The comparison is made on a couple of criteria including:

- Approach .
- Algorithm used .
- Type of algorithm .
- Dataset used.
- Application Area .
- Framework.
- Results.

ML Paradigm	**Approach**	*Used Algorithm*	*Type of algorithm*	*Dataset used*	*Results*
Unsupervised	Cheney et al. 2020	Krimp CompreX AVC AVC*	Machine learning	Pang DARPA	Best Algorithm AVC
Unsupervised	Benabderrahmane et al. 2021	VR-ARM VF-RAM	Deep learning	DARPA	VR-ARM
Unsupervised	Berrada et al. 2020	FPOF OD OC3 CompreX AVF	Deep learning	DARPA	Best Algorithm AVF
Unsupervised	Akoglu et al. 2012	COMPREX KRIMP	Machine learning	Complex Dataset	Best Algorithm COMPREX
Unsupervised	Syarif, Prugel-Bennett, and Wills 2012	Clustering (k-means, k-medoids, EM, outlier detection algorithm)	Machine learning	NSL-KDD	Average Accuracy =71.62%
Unsupervised	Aslahi-Shahri et al. 2016	GA SVM	Machine learning	DARPA KDD Cup99	precision =0.941
Supervised	Chalapathy, Menon, and Chawla 2018	OC-NN	Deep learning	Complex Dataset	/
Supervised	Chu, Lin, and Chang 2019	SVM Naive Bayes Decision tree , MLP	Machine learning	NSL-KDD	Best Algorithm SVM
Supervised	Wang et al. 2017	CNN	Deep learning	USTC-TFC2016 USTC-TK201	accuracy =99.41%
Supervised	Andropov et al. 2017	ANN	Deep learning	USTC-TFC2016 USTC-TK201	accuracy =80%
Supervised	Vargas-Muñoz et al. 2018	Classification (Bayesian)	Machine learning	Real traffic, UNB ISCX IDS 2012, UAN W32.Worms	Average Accuracy =99%(UAN) Average Accuracy =99.3%(UAB)
Supervised	Shen and Chow 2020	ANN	Machine learning	ISP	Average Accuracy =87.71%

ML Paradigm	**Approach**	*Used algorithm*	*Type of algorithm*	*Dataset used*	*Results*
Semi-Supervised	Demertzis, Iliadis, and Bougoudis 2020	Gryphon (eSNN eSNN-OCC)	Deep learning	water tower gas_dataset electric_dataset	F_score=0.98 F_score=0.97 F_score=0.96
Semi-Supervised	Barceló-Rico, Esparcia-Alcázar, and Villalón-Huerta 2016	GP CART RF SVM	Machine learning	APT-infected HTTP	/
Semi-Supervised	Chen et al. 2018	AE CAE	Deep learning	NSL-KDD	Average Accuracy =95.85%
Semi-Supervised	Chahal and Kaur 2016	K-mean aSVM	Machine learning	NSL-KDD	Average Accuracy =98.47%

Table 5.14: Comparative analysis

5.7 Major milestones from the approaches :

Through a brief review of anomaly based intrusion detection system approaches this section provides a comparison of these approaches by giving the advantages and short comings of each approach. Firstly, Based on the different approaches that we have previously reviewed, we can emphasize that that VF-ARM, AVF , AVC and COMPREX gave the best results in the Unsupervised learning and the SVM in the supervised learning .

All the approaches used the same following data (NSL KDD , DARPA , BSP)which aimed in one topic , which is the Cybersecurity.

Secondly, from the results of the previous table 5.6, we noticed that the best results for precision,F-score and Recall were achieved by applying the SVM and CNN.

Finally ,After reviewing several studies realized with the aim of detecting anomaly , we observed that using the autoencoders to detect APTs attack are rares , or even non-existent. So we decided to talk about it and implement it in our engineer project .

5.8 Conclusion :

Because of its relevance in safeguarding systems and reducing the danger of being compromised, APTs anomaly detection has become a major research topic, as we illustrated in this chapter. Over the years, a large variety of anomaly-based intrusion detection algorithms and methodologies have been created by researchers. Since no solution guarantees total protection against Systems attacks and abnormalities, each of these solutions has its own characteristics, benefits, and downsides. Despite this impressive improvement, there are still several chances to improve the state-of-the-art in identifying and preventing cybercrime.resolving network abnormalities

Part IV

Conclusion

Anomalies have always existed to endanger systems and enterprises' information assets, ranging from simple hardware failure and bugs to complex and sophisticated cyber-attacks. As a result, seeking immunity through advanced intrusion detection systems capable of detecting new attack techniques becomes increasingly crucial.

The goal of this thesis was to highlight the anomaly detection problem in terms of several connected elements. It also aimed to examine the current state-of-the-art in modern anomaly-based network intrusion detection systems; as a result, many papers were reviewed in this study in order to provide an overview of what has been done in the anomaly detection domain, as well as what could be improved in order to ensure better protection and enhanced immunity against current threats.

This survey also included a brief explanation of the various current datasets and their categorization. Furthermore, it highlighted the significant gap identified in the lack of standard and current intrusion datasets, emphasizing the importance of generating datasets containing a set of attacked data based on the reactions of these attacks on some features.

This survey tries to provide a structured and comprehensive overview of the research on anomaly detection. We hope that it facilitates a better understanding of the different directions in which research has been done on this topic, and how techniques developed in one area can be applied in domains for which they were not intended to begin with.

BIBLIOGRAPHY

[AC20] S Abirami and P Chitra. "Energy-efficient edge based real-time healthcare support system." In: *Advances in Computers*. Vol. 117. 1. Elsevier, 2020, pp. 339–368.

[Aga19] Rahul Agarwal. *The 5 Classification Evaluation metrics every Data Scientist must know*. https://towardsdatascience.com/the-5-classification-evaluation-metrics-you-must-know-aa97784ff226. 2019.

[AC14] Nam-Uk Ahn Sung-Hwan Kimć and Tai-Myoung Chung ć. "Big data analysis system concept for detecting unknown attacks." In: (2014), pp. 269–272.

[Ako+12] Leman Akoglu et al. "Fast and reliable anomaly detection in categorical data." In: (2012), pp. 415–424.

[And+17] Sergey Andropov et al. "Network anomaly detection using artificial neural networks." In: (2017), pp. 26–31.

[Asl+16] BM Aslahi-Shahri et al. "A hybrid method consisting of GA and SVM for intrusion detection system." In: *Neural computing and applications* 27.6 (2016), pp. 1669–1676.

[BEV16] Fàtima Barceló-Rico, Anna I Esparcia-Alcázar, and Antonio Villalón-Huerta. "Semi-supervised classification system for the detection of advanced persistent threats." In: (2016), pp. 225–248.

[Ben+21] Sidahmed Benabderrahmane et al. "A rule mining-based advanced persistent threats detection system." In: *arXiv preprint arXiv:2105.10053* (2021).

[Ber+20] Ghita Berrada et al. "A baseline for unsupervised advanced persistent threat detection in system-level provenance." In: *Future Generation Computer Systems* 108 (2020), pp. 401–413.

[Bro19] Jason Brownlee. *14 Different Types of Learning in Machine Learning*. https://machinelearningmastery.com/types-of-learning-in-machine-learning/. Accessed on 2022-02-12. 2019.

[Bru21] Kate Brush. *deep learning?* https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network. Accessed on 2022-02-12. Mar. 2021.

[CK16] Jasmeen K Chahal and Amanjot Kaur. "A hybrid approach based on classification and clustering for intrusion detection system." In: *International Journal of Mathematical Sciences & Computing* 2.4 (2016), pp. 34–40.

[CMC18] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. "Anomaly detection using one-class neural networks." In: *arXiv preprint arXiv:1802.06360* (2018).

[Cha00] Varun Chandola. "Anomaly detection for symbolic sequences and time series data." In: University of Minnesota, 200.

[CBK09a] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey." In: *ACM Comput. Surv.* 41 (2009), pp. 15–15. DOI: 10.1145/1541880.1541882.

[CBK09b] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[Che+18] Zhaomin Chen et al. "Autoencoder-based network anomaly detection." In: (2018), pp. 1–5.

[Che+20] James Cheney et al. "Categorical anomaly detection in heterogeneous data using minimum description length clustering." In: *arXiv preprint arXiv:2006.07916* (2020).

[CLC19] Wen-Lin Chu, Chih-Jer Lin, and Ke-Neng Chang. "Detection and classification of advanced persistent threats and attacks using the support vector machine." In: *Applied Sciences* 9.21 (2019), p. 4579.

[Coh17] David Cohn. "Active Learning." In: *Encyclopedia of Machine Learning and Data Mining*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2017, pp. 9–14. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_916. URL: https://doi.org/10.1007/978-1-4899-7687-1_916.

[] *Daniel Johnson*. en-US. URL: https://www.guru99.com/daniel-johnson-2 (visited on 2022).

[DIB20] Konstantinos Demertzis, Lazaros Iliadis, and Ilias Bougoudis. "Gryphon: a semi-supervised anomaly detection system based on one-class evolving spiking neural network." In: *Neural Computing and Applications* 32.9 (2020), pp. 4303–4314.

[Dem+14] Howard B Demuth et al. "Neural network design." In: Martin Hagan, 2014.

[Gér19] Aurélien Géron. "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow." In: July 2019.

[Gol+00] Ary L Goldberger et al. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals." In: *circulation* 101.23 (2000), e215–e220.

[Goy21] Kechit Goyal. *6 Types of Supervised Learning You Must Know About in 2022*. https://www.upgrad.com/blog/types-of-supervised-learning/. Accessed on 2022-02-12. 2021.

[Hol+18] Andreas Holzinger et al. "Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable AI." In: *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. 2018, pp. 1–8. DOI: 10.1007/978-3-319-99740-7/_1.

[Ian16] Aaron Courville Ian Goodfellow Yoshua Bengio. "Deep Learning." In: http://www.deeplearningbook.org. MIT Press, 2016.

[Al-18] Layth Al-Jaberi. “Chapter Three .”

$^{Artificial Neural Networks ARTIFICIAL NEURAL NETWORKS 3.1 General}$

In: (Sept. 2018).

[Jai+21] Ashish Jaiswal et al. “A Survey on Contrastive Self-Supervised Learning.” In: *Technologies* 9.1 (2021). ISSN: 2227-7080. DOI: 10.3390/technologies9010002. URL: https://www.mdpi.com/2227-7080/9/1/2.

[Joh22] Daniel Johnson. *Unsupervised Machine Learning: Algorithms, Types with Example*. https://www.guru99.com/unsupervised-machine-learning.html. Accessed on 2022-05-20. 2022.

[Khr+19] Ansam Khraisat et al. “Survey of intrusion detection systems: techniques, datasets and challenges.” In: *Cybersecurity* 2.1 (2019), pp. 1–22.

[LW12] Qiong Liu and Ying Wu. “Supervised Learning.” In: (Jan. 2012). DOI: 10.1007/978-1-4419-1428-6_451.

[mad22] madarsh986. *Inductive Learning Algorithm*. https://www.geeksforgeeks.org/inductive-learning-algorithm/. Accessed on 2022-05-20. 2022.

[MMP97] Kishan Mehrotra, Chilukuri Mohan, and Sanjay Preface. “Elements of Artificial Neural Nets.” In: Jan. 1997.

[Mis18] Aditya Mishra. *Metrics to Evaluate your Machine Learning Algorithm*. https://towardsdatascience.com/metrics-to-evaluate-your-machinelearning-algorithm-f10ba6e38234. 2018.

[Mus21] Mustapha. *Optimizers in Deep Learning*. https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0. Accessed on 2022-04-14. 2021.

[Mwi+19] Henry Mwiki et al. “Analysis and triage of advanced hacking groups targeting western countries critical national infrastructure: APT28, RED October, and Regin.” In: *Critical infrastructure security and resilience*. Springer, 2019, pp. 221–244.

[SW10] “Deductive Learning.” In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 267–267. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_206. URL: https://doi.org/10.1007/978-0-387-30164-8_206.

[SSA17] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. “Activation functions in neural networks.” In: *towards data science* 6.12 (2017), pp. 310–316.

[SC20] Ao Shen and Kam Pui Chow. “Time and Location Topic Model for analyzing Lihkg forum data.” In: (2020), pp. 32–37.

[Sne20] Hirendra Hazare Sneha Sakhareć. “Big Data System for Detecting Unknown Attacks.” In: (2020).

[SPW12] Iwan Syarif, Adam Prugel-Bennett, and Gary Wills. “Unsupervised clustering approach for network anomaly detection.” In: (2012), pp. 135–145.

[Tav+09] Mahbod Tavallaee et al. “A detailed analysis of the KDD CUP 99 data set.” In: (2009), pp. 1–6.

[VH20] Jesper E Van Engelen and Holger H Hoos. “A survey on semi-supervised learning.” In: *Machine Learning* 109.2 (2020), pp. 373–440.

[Var+18] MJ Vargas-Muñoz et al. “Classification of network anomalies in flow level network traffic using Bayesian networks.” In: (2018), pp. 238–243.

[VD15a] J Vukalović and Damir Delija. “Advanced persistent threats-detection and defense.” In: (2015), pp. 1324–1330.

[VD15b] J Vukalović and Damir Delija. “Advanced persistent threats-detection and defense.” In: (2015), pp. 1324–1330.

[Wan+17] Wei Wang et al. “Malware traffic classification using convolutional neural network for representation learning.” In: (2017), pp. 712–717.