الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Mémoire de Master

Spécialité : Sécurité Informatique et Cryptographie

Thème

A Residual Learning based Network Intrusion Detection System

Présenté par :

- **Abdelouahab Oussama Hassan**
- **Boudaoud Abderrahmane**

Dirigé par :

- **BENAMARA Djilali**

Promotion 2021 - 2022

Acknowledgements

First of all, we thank Allah who by his good grace we were able to finish this work.

Our sincere gratitude to ***Dr.Benamara Djillali***, who guided and

mentored us through the process of making this work a reality and to whom we are indebted to.

we would like to thank the members of the jury for doing us the honour of evaluating

our humble work.

Our undying gratitude to faculty members, teachers and our classmates for being

delightful companions on our journey to learn.

We want to express our thanks to our friends and family memberfor being

the best support system we could have wished for.

Dedication

To my parents, my sisters and my brother Mohamed

My friends specially Yaçine, Ilyes and Abderrahmane. To all the teacher who have helped me cultivate knowledge the entirety of my school career.

Abderrahmane

To my parents, who raised, teached, encouraged and supported me throughout all my life.

To my brothers and beloved sister.

To my friends Ilyes, Mohamed, Moussa and Yacine.

Hassan

Contents

List of Figures

List of Tables

Abréviation

- **AI:** Artificial Intelligence.
- **ML:** Machine Learning.
- **SVM:** Support Vector Machines.
- **SSL:** Semi-supervised learning .
- **IDS:** Intrusion Detection System.
- **NIDS:** Network based Intrusion Detection System.
- **HIDS:** HOST based Intrusion Detection System.
- **DL:** Deep Learning
- **ANN:** Artificial Neural Network.
- **CNN:** Convolutional Neural Network.
- **ResNet:** residual network.
- **SGD:** Stochastic Gradient Descent

General Introduction

Nowadays, networks and computer systems have became the equivalent of vital organs for all and any functioning entities no matter their field (Universities, enterprises, financial institutions, military and different services ... etc).

Unfortunately just like an organ is exposed to sickness, information technology system is vulnerable to attacks that take advantage of weaknesses in the system architecture. These attacks are evolving rapidly and they aim to enslave the host system making it of servitude to a third party. To avoid this horrendous outcome companies put in a lot of effort to secure and protect computer networks, hoping to make them impenetrable.

Their arsenal mainly relies on Intrusion Detection Systems, they seek to detect attacks and capture intrusions inside the local network. Furthermore they take notice in analysing and hypothesizing different attack scenarios based on pre-obtained data. Its process enables it to reinforce its defences and cutting off the attacks root to stem. Being cautious as in early detecting an attack scenarios making it possible to quickly stop their development and consequently avoid more serious damage.

Despite having decades of development, Intrusion Detection Systems still have room for improvement in regards of detection accuracy, reducing false positive rate and detecting unknown attacks. Capitalizing on the rapid development of Machine Leaning methods, researchers sought out to enrol them in cyber security field, because and given the fact that they are able to differentiate between normal and malicious data while also being capable of detecting new forms of attacks. An inquiry about Deep Learning will lead to the understanding that it is a branch of machine learning which is able to outperform all other branches. It is capable of handling large volumes of data, the complexity of Deep Learning having multiple hidden layers enables it reach heights that are difficult for traditional machine learning models.

0.1 Objectif

In this project we sought out to fully explore machine learning and gain an absolute comprehension of its different types then we embarked on a quest to discover what are Intrusion Detection systems, their types and detection methods.
Furthermore we intended to gain perspective on Residual Learning and clarify how it has adopted deep learning to take its performance to new heights.

0.2 Chapter description :

0.2.1 Chapter 1 :

The first chapter introduces Machine Learning field while detailing its types to a tee. Furthermore it explores Intrusion Detection Systems, charts their types and detection methods.

0.2.2 Chapter 2

The second chapter aims to highlight deep learning, its application and neural networks, then we brought forward Residual Learning, navigating through its meaning and its architecture.

0.2.3 Chapter 3 :

The third chapter shines a light on our suggested model, tools that served to bring our research to its conclusion and the results we obtained.

Chapter01

Machine Learning

1 Machine Learning

1.1 Introduction

Machine learning can be considered a branch of artificial intelligence. Indeed, a system that can be incapacitated to learn can hardly be considered intelligent. The ability to learn and draw from one's experiences is of course essential to a system designed to adapt to the changing environment surrounding it.

A general and high-level definition of AI would be as the theory and development of computer systems that perform tasks that augment for human intelligence, such as perceiving, classifying, learning, abstracting, reasoning, and/or acting, so where is the place of ML ?, what does ML provide to this field?. It is due to machine learning we're having these smart games, intelligent voice recognition software, online fraud detection, and possibly in the future of Automotive self-driving vehicles.

So what is Machine Learning ?

The first and early definition of ML was introduced by Arthur Samuel in 1959 on which he said :

"machine learning is the field of study that gives computers the ability to learn without being explicitly programmed".

But the idea is, how can we have a computer learn without being explicitly programmed ?, And one way to think about this is to think about the difference between how we would normally program? and what we would like from a machine learning algorithm ?.

Machine learning lies mainly in the ability to develop programs that use data through the process of converting it into some form of information, then passing said information to a series of algorithms in a way that they discover themselves using that specific data.

The fact that we can deduce new predictions, actions, facts, learning from previous experiences, gradually improves the performance of predictive models, and make data-driven decisions by observing data, however capturing the knowledge in data is the real process of learning.

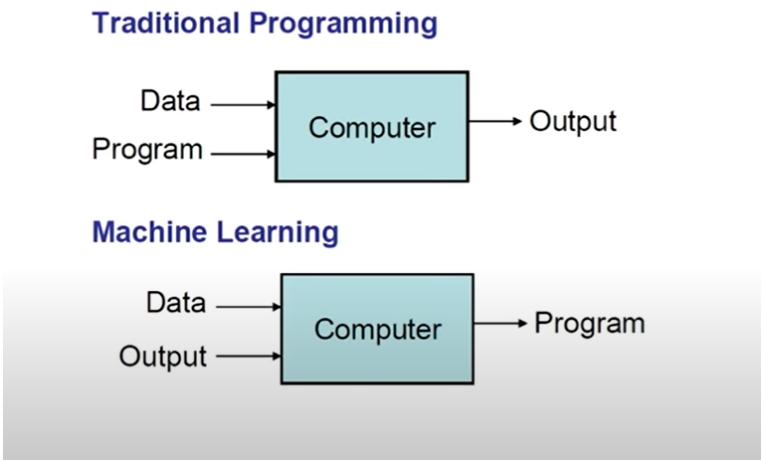


Figure 1: difference between TP and ML[1].

1.2 Types of ML :

Machine Learning is a broad field, where exists different types of machine learning systems we list in this section the major four types of machine learning. Machine Learning systems can be classified according to the amount and type of supervision they get during training.

There are four major categories: supervised learning, unsupervised learning, semi-supervised learning and Reinforcement Learning [2]

1.2.1 Supervised learning :

This is the most common category. The main purpose of supervised learning is to make predictions for new input data after passing the algorithm the training data including the desired output results, called labels.

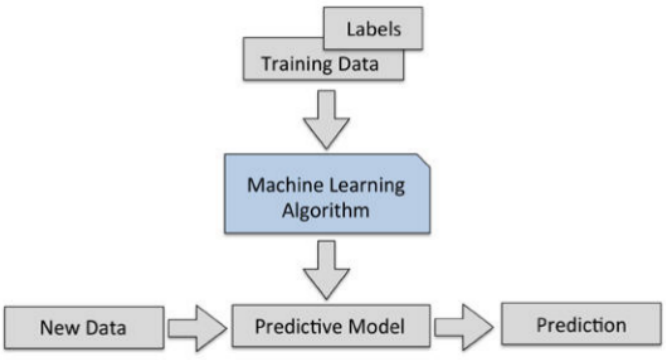


Figure 2: A resume of supervised learning category

[3]

A typical supervised learning task would be the robust e-mail spam filters we are enjoying nowadays, so we can train a model using a supervised machine learning algorithm on a corpus of labeled e-mail, e-mail that are correctly marked as spam or not-spam, to predict whether a new e-mail belongs to either of the two categories.

A supervised learning task with discrete class labels, such as in the e-mail spam-filtering example [3] leads us to distinguish between two subcategories of supervised learning : Classification and regression.

- **Classification:**

  Classification is the most widely used technique where we try to predict whether the input data belongs to a specific data or not, just like the previous spam filtering example.

  Classification is a data mining (machine learning) approach that used to forecast group membership for data instances [4]

  Classification is an admired task in machine learning, especially in future plan and knowledge discovery. Classification is categorized as one of the supreme studied problems by researchers of the machine learning and data mining fields [5].

- **Regression :**

  The second subcategory of supervised learning would be Regression, where output values take continuous outcomes it does not take class labels like the previous subcategory, it is used to perform predictive analysis and find a linear correspondence between one or more predictors(data points).

  Regression is a technique used for two theories. First, regression analyses is usually used for forecasting and prediction, in which their application has major overlaps with the area of machine learning. Second, regression analysis can be used in some cases to determine causal relations between the independent and dependent variables. More importantly, regressions alone show only relations between a dependent variable and a fixed data-set collection of different variables. [6].

  Figure 03 illustrates the concept of linear regression. Given a predictor variable x and a response variable y, we fit a straight line to this data that minimizes the distance—most commonly the average squared distance—between the sample points and the fitted line. We can now use the intercept and slope learned from this data to predict the outcome variable of new data[3]:

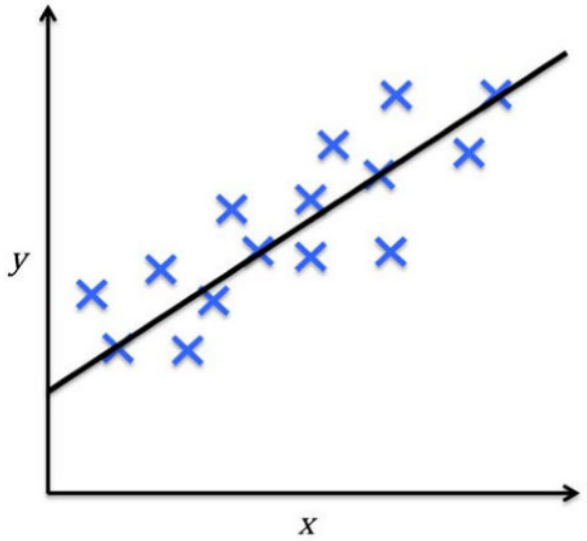


Figure 3: Regression[3]

- **Support Vector Machines (SVM) :**

  SVMs are one of the most mysterious methods in Machine Learning, the main idea behind them is to try to find a linear separator, that separates our input data classes. The shortest distance between the data classes and the separator is called margin. The SVM technique is a classifier that finds a hyperplane that correctly separates two classes with a maximum margin. Figure 04 shows a separating hyperplane corresponding to a hard-margin SVM (also called a linear SVM) [7].

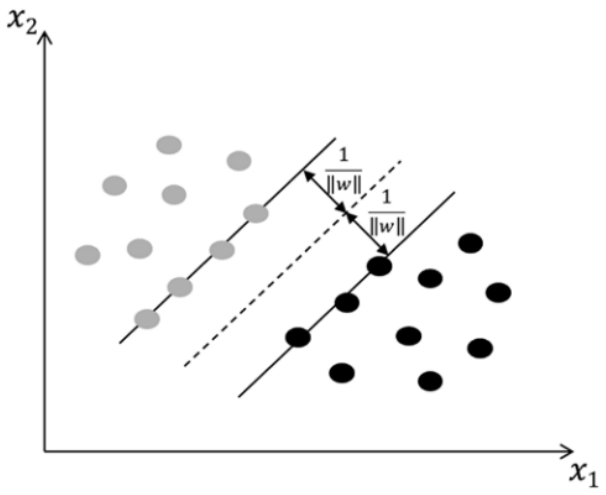


Figure 4: Margin separation

So when we have two categories, but no obvious linear classifier that separates them in a a logical way, Support Vector Machines work by moving the data into a relatively high dimensional space and finding a relatively high dimensional Support Vector Classifier that can effectively classify the input data.

1.2.2 Unsupervised learning :

It means no supervision, the algorithm gets only inputs and no labels (outputs). Unsupervised learning depends essentially on Clustering (Figure 05), where the algorithm tries to find similarities between inputs, and categorize them together, it is all being done through observation.

Clustering refers to a very broad set of techniques for finding subgroups, or clustering clusters, in a data-set.

When we cluster the observations of a data-set, we seek to partition them into distinct groups so that the observations within each group are quite similar to one another, while observations in different groups are quite different from one another. Of course, to make this concrete, we must define what it means for two or more observations to be similar or different [8].

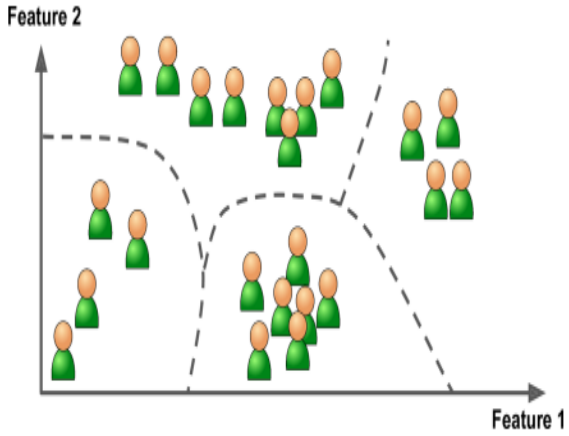


Figure 5: Clustering [2]

1.2.3 Semi-supervised learning :

Semi-supervised learning uses a combination between supervised learning and unsupervised learning techniques and that is because in a scenario where we would make use of semi-supervised learning we would have a combination of both labelled and unlabelled data which would boost the accuracy of the ML models.

The main objective of Semi-supervised learning is to overcome the drawbacks of both supervised and unsupervised learning.

Supervised learning requires huge amount of training data to classify the test data, which is a cost effective and time consuming process. On the other hand, unsupervised learning does not require any labeled data, which clusters the data based on similarity in the data points by using either clustering or maximum likelihood approach. The main downfall of this approach, it can't cluster an unknown data accurately.

To overcome these issues, Semi-supervised learning has been proposed by research community, which can learn with small amount of training data can label the unknown (or) test data. SSL builds a model with few labeled patterns as training data and treats the rest of the patterns as test data [9].

1.2.4 Reinforcement learning :

Reinforcement learning fills the gap between supervised learning and unsupervised learning, it is a strategy where the machine acts as an agent that learns from its environment by letting it come up with its own data and examples rather than being fed to it, in an interactive way until it determines the suitable behaviours.

In reinforcement learning, the algorithm gets feedback in the form of a reward; about how well it is doing.In contrast to supervised learning, where the algorithm is 'taught' the correct answer, the reward function evaluates the current solution, but does not suggest how to improve it.

Just to make the situation a little more difficult, we need to think about the possibility that the reward can be delayed, which means that you don't actually get the reward instantly, but for a delayed period of time. (For example, think about a robot that is learning to traverse a maze.

It does not know whether it has found the centre of the maze until it gets there, and it does not get the reward until it reaches the centre of the maze.) therefore, we need to allow for rewards that don't appear until long after the relevant actions have been taken.

Sometimes we think of the immediate reward and the total expected reward into the future. Once the algorithm has decided on the reward, it needs to choose the action that should be performed in the current state. This is known as the policy. This is done based on some combination of exploration and exploitation (remember, reinforcement learning is basically a search method), which in this case means deciding whether to take the action that gave the highest reward last time we were in this state, or trying out a different action in the hope of finding something even better [10].

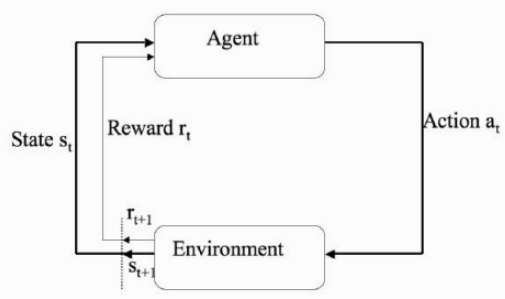


Figure 6: The Reinforcement Learning cycle[10].

The following table shows advantages and disadvantages of the four types of machine learning :

Type	**Advantages**	**Disadvantages**
Supervised Learning	- the algorithm's ability to achieve good results with a small number of variables. - the simplicity, the speed and easy training of large data volumes	- failure to predict rare events and it is possible to overfit - the requirement of multiple training examples and difficulty of result interpretation
Unsupervised Learning	- simple, flexible, efficient and easy to manage - good for segmentation and easy to interpret	- does not allow to build the most optimal set of clusters. - the order and the noise of the data may affect the results.
Semi-supervised Learning	- intuitive and very easy to understand. It produces very good results - low sensitivity and the possibility to apply almost all existing classifiers	- the added unlabeled data contaminates the original labelled data. - the period sometimes required to train its algorithm
Reinforcement Learning	- fast and does not require adaptation and it can learn successfully from incomplete data - works well with complex systems; it can maintain good control and allows sensitivity analysis and optimization of the real system	- Number of steps can sometimes be limited.

Table 1: major advantages and disadvantages of different ML methods [41].

machine learning

1.3 Intrusion Detection System (IDS)

1.3.1 Introduction

Over the past two or three years, many people thought that if they wanted to connect to the Internet, only a firewall was needed to complete the security puzzle. But this is not always true With the arrival of the big data era, approaches to network attack are being updated in a daily basis.

New network intrusions have shown a trend of intelligentization and complication. It is difficult for traditional anomaly detection technologies to take effect on new network intrusions and deliver a satisfying result[11].

Before we get to what is IDS, we need to clarify these main terms, system, intrusion and audit.

The term system(a.k.a target system) is used to denote the information system being monitored by the intrusion-detection system. It can be a workstation, a network element, a server, a mainframe, a firewall, a web server, etc..[12].

As for the term intrusion, it describes any use of a computer system for any of its intended purposes, usually due to the acquisition of privileges in an illegitimate manner. For the term Audit it denotes information provided by a system concerning its inner workings and behaviour[12].

1.3.2 Definition:

An intrusion-detection system acquires information about an information system to perform a diagnosis on the security status of the latter.

The goal is to discover breaches of security, attempted breaches, or open vulnerabilities that could lead to potential breaches. A typical intrusion-detection system is shown in Figure

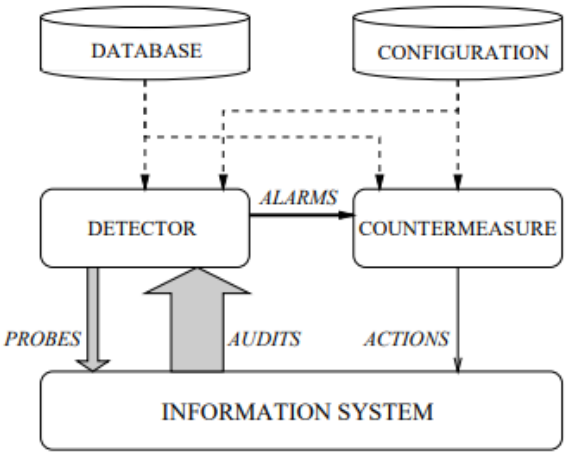


Figure 7: Very simple intrusion-detection system[12].

In other words, IDS is a mechanism designed to detect abnormal or suspicious activity on the scanned target (a network or host). Therefor it makes it possible to have a preventive action on the risks of intrusion, in order to detect attacks that a system may suffer, it is necessary to have specialized software whose role will be to monitor the data that passes through this system and which would be able to react if data seems suspicious.

Primary criterions of measurement for IDS are as follow :

- TRUE POSITIVE: legitimate attack (IDS gives alarm).
- FALSE POSITIVE: no attack (IDS gives alarm).
- FALSE NEGATIVE: legitimate attack (IDS gives no alarm)
- TRUE NEGATIVE: no attack (IDS gives no alarm) [13].

1.4 Types of intrusion detection systems :

Because of the diversity of attacks that hackers implement, intrusion detection must be done at several levels. So there are different types of IDS's :

1.4.1 Network based Intrusion Detection System (NIDS):

A NIDS listens for all network traffic, then analyses it and generates alerts if packets seem dangerous. NIDS monitor the whole traffic of the network from which the hosts are connected, it obtains data from them and make their decisions. NIDS is cost effective and gives immediate real time detection of the network attacks so it reduces and decreases the chance of the damages of the network because of the intrusion activities[13]. NIDS are the most interesting and useful IDS because of the ubiquity of networks in our daily lives.

machine learning

1.4.2 HOST based Intrusion Detection System (HIDS):

A HIDS is based on a single machine, this time no longer analysing network traffic but the activity happening on this machine.

HIDS attempts to identify unauthorized, illicit, and anomalous behavior on a specific device. HIDS generally involves an agent installed on each system, monitoring and remotely alerting on local OS and application activity [14]. Some common abilities of HIDS system include log analysis, event correlation, integrity checking, policy enforcement, rootkit detection and alerting [15].

A HIDS needs a healthy system to verify the integrity of the data, if the system has been compromised by a hacker, HIDS will no longer be effective, It needs a healthy system to verify the integrity of the data, if the system has been compromised by a hacker, HIDS will no longer be effective.

1.4.3 Hybrid Intrusion Detection System :

Generally used in a decentralized environment, they make it possible to gather information from various probes placed on the network. An Hybrid IDS captures data from both host and network, which can be further analyzed for possible intrusions.
Since these systems will work for both host and network they prove to be more efficient than traditional IDS.

machine learning

An Hybrid IDS takes advantage from both the approaches, and develops an IDS that works for both a host and a network.

These systems are the future IDS A lot of research is currently is carried out to overcome the drawbacks of HIDS and NIDS, and develop an Hybrid system that work for both host and network[15].

1.5 Efficiency of intrusion-detection systems :

The main three measures to evaluate the efficiency of an IDS have been proposed by Porras et al. in [16] :

- **Accuracy:** accuracy is when there is no false alarm. All attacks are detected. Accuracy decreases when the IDS gives an alarm to a legitimate behaviour (False Positive).
- **Performance:** performance is measured by the rate at which events are processed, a real-time detection would not be possible with a low rate.
- **Completeness:** this measure is the most difficult, because it deals with the ability of an IDS to detect all attacks which is impossible considering the fact that we can not have a global knowledge of the attacks.

  Debar et al. in [17] also added the following two measures :

- **Fault tolerance:** An intrusion-detection system should itself be resistant to attacks, especially denial-of-service attacks.

machine learning

This is particularly important because most intrusion-detection systems run above commercially available operating systems or hardware, which are known to be vulnerable to attacks

- **Timeliness:**An intrusion-detection system has to perform and propagate its analysis as quickly as possible to enable the security officer to react before much damage has been done, and also to prevent the attacker from subverting the audit source or the intrusion-detection system itself. This implies more than the measure of performance because it not only encompasses the intrinsic processing speed of the intrusion-detection system, but also the time required to propagate the information and react to it.

1.6 Detection Methods:

There are two complementary trends in intrusion detection: to use the knowledge accumulated about attacks and look for evidence of the exploitation of these attacks, and to build a reference model of the usual behavior of the information system being monitored and look for deviations from the observed usage.

1.6.1 Knowledge-based intrusion detection:

Knowledge-based intrusion-detection techniques apply the knowledge accumulated about specific attacks and system vulnerabilities.

The intrusion-detection system contains information about these vulnerabilities and looks for attempts to exploit them.

When such an attempt is detected, an alarm is raised. In other words, any action that is not explicitly recognized as an attack is considered acceptable. Therefore, the accuracy of knowledge-based intrusion-detection systems is considered good. However, their completeness depends on the regular update of knowledge about attacks Advantages of the knowledge-based approaches are that they have, in theory, very low false-alarm rates, and that the contextual analysis proposed by the intrusion-detection system is detailed, making it easier for the security officer using this intrusion-detection system to understand the problem and to take preventive or corrective action.

Drawbacks include the difficulty of gathering the required information on the known attacks and keeping it up to date with new vulnerabilities and environments.

Maintenance of the knowledge base of the intrusion detection system requires careful analysis of each vulnerability and is therefore a time-consuming task. Knowledge-based approaches also have to face the generalization issue. Knowledge about attacks strongly depends on the operating system, version, platform, and application. The resulting intrusion-detection system is therefore closely tied to a given environment. Also, detection of insider attacks involving an abuse of privileges is deemed more difficult because no vulnerability is actually exploited by the attacker[12].

1.6.2 Behavior-based intrusion detection:

Behavior-based intrusion-detection techniques assume that an intrusion can be detected by observing a deviation from the normal or expected behavior of the system or the users. The model of normal or valid behavior is extracted from reference information collected by various means. The intrusion-detection system later compares this model with the current activity, When a deviation is observed, an alarm is generated.

In other words, anything that does not correspond to a previously learned behavior is considered intrusive. Therefore, the intrusion-detection system might be complete, but its accuracy is a difficult issue. Advantages of behavior-based approaches are that they can detect attempts to exploit new and unforeseen vulnerabilities. They can even contribute to the (partially) automatic discovery of these new attacks. They are less dependent on operating-system-specific mechanisms.

They also help detect "abuse-of-privilege"-type attacks that do not actually involve exploiting any security vulnerability.

The high false-alarm rate is generally cited as the main drawback of behavior-based techniques because not the entire scope of the behavior of an information system may be covered during the learning phase. Also, behavior can change over time, introducing the need for periodic on-line retraining of the behavior profile, resulting either in unavailability of the intrusion-detection system or in additional false alarms. The information system can undergo attacks at the same time the intrusion-detection system is learning the behavior.

As a result, the behavior profile will contain intrusive behavior, which is not detected as anomalous[12].
A better explanation of these two detection methods in the table below:

	Behavior-based intrusion detection	Knowledge-based intrusion detection :
Advantages	Detection of new attacks	Easy explanation of attack scenarios - Minimal false positive generation
Inconvenient	Inaccurate learning: if it is based on data that is supposed to be normal but contains unknown attacks	- Failure to detect new attacks
	- Evolution of normal behavior which requires adaptive learning risk of progressive learning initiated by an intruder	-Expertise required to effectively construct attack signatures
	- huge generation of false positives	- Continuous updating of the rule base
	- Crucial selection of useful parameters during the learning phase since excessive parameters represent noise while those less degrade the quality of detection	- - Fast increase in the number of rules that generally lack abstraction
	- Difficult definition of the exact thresholds of anomaly	- Long delays between the discovery of attacks and the definition of signatures
	- Difficult to set the durations necessary for learning	

Table 2: Comparison of the two intrusion detection Methods[18].

1.7 Conclusion:

This chapter has been devoted to the presentation of Machine Learning and its types. Then it explored Intrusion Detection Systems, their types and different detection methods. The next chapter will discuss in details Deep Learning, neural networks and furthermore Residual Learning.

Chapter02

Deep Learning

2 Deep Learning

2.1 Introduction

Nowadays, DL is in the center of attention since its realization is much more important than any other machine learning algorithm in such complex tasks, for example, we mark the following:

- Image processing and object recognition in [19] which show us a progress using deep convolutional networks for object recognition, and the adoption of deep learning by the computer vision community.

- Voice recognition and signal processing in [20] which present the results obtained in the phonetic classification for the automatic recognition of speech as the first industrial application of deep learning.D.L is one of the reasons driving recent AI movements and advancements, and this is the main cause that makes one think: finally, there is a possibility for the AI to become more realistic.

So what is DL?

According to founders Yann LeCun, Yoshua Bengio and Geoffrey Hinton in [21]:

"Deep learning enables computer models composed of multiple layers of processing to learn representations of data with multiple levels of abstraction".

Another definition by the authors in [22] :

"Deep learning is a class of machine learning techniques, where information is processed in hierarchical layers to understand the representations and characteristics of data in increasing levels of complexity health."

In other words, DL is a subset of ML methodologies and techniques that use the artificial neural network (ANN). It is the adaptation of neural networks that imitates the structure of the human brain. The strength of DL lies in the fact that the machine can extract characteristics and learn on their own, independent of the intervention of an expert. It has been applied in many different fields (image processing, text, speech and videos). The success of DL belongs to the availability of more training data [23].

The following figure represents a summary of the progress of AI from its beginnings until it has been applied in many different fields (image processing, text, speech and videos). The success of DL belongs to the availability of more training data [23].

The following figure represents a summary of the progress of AI from its beginnings until the appearance of DL.

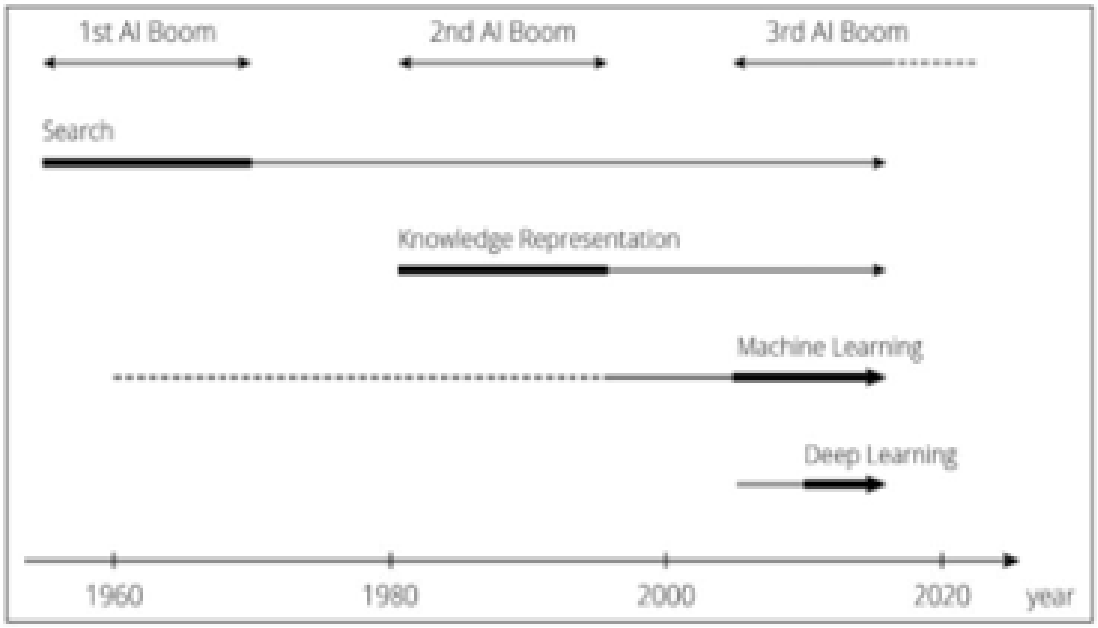


Figure 8: Advances in Artificial Intelligence [24]

In [68], an article represents an overview of the historical context of DL, it presents the major steps that lead to what we have now.

These steps are summarized in the The following table:

year	contribution
1763	Bayesian Networks
1913	Markov Chains
1950	Turing Test
1951	First Neural Network Machine
1958	Perceptron
1980	CNN
1982	RNN
1986	Backpropagation
1989	Reinforcement Learning
1995	SVM
1997	LSTM
1998	LeNet
2012	AlexNet
2014	GoogleNet, VGG net
2015	ResNet

Table 3: Time line of Deep Learning [42]

The relationship between the three concepts AI, ML and DL is summarized by the authors [25] in the following figure:

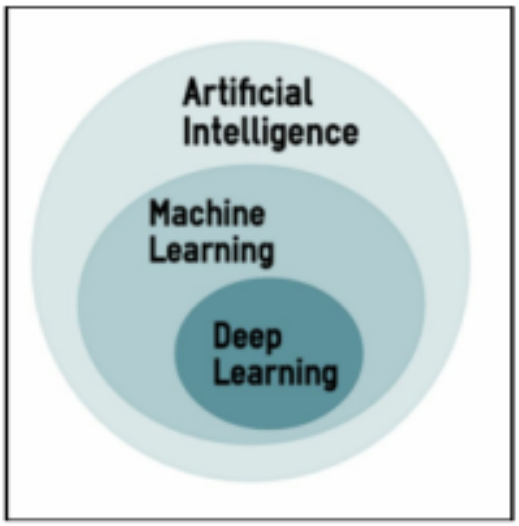


Figure 9: The relationship between the 3 concepts [25]

2.2 The applications of Deep Learning :

It has found a footing in numerous fields to name a few :

2.2.1 Facial recognition

The eyes, the nose, the mouth, just as many characteristics that a DL algorithm will learn to detect in a photo. It's going to be first place to give a certain number of images to the algorithm, then due to training intensity, the algorithm will be able to detect a face on an image.

2.2.2 Natural language processing

The automatic processing of Natural language is another application of DL. Its purpose is to extract the meaning of words, or even sentences.

The algorithm goes by example understand what is said in a Google review, or will communicate with people via chatbots. Automatic text reading and analysis is also a fields of application of the DL with Topic Modeling: such text addresses such a subject.

2.2.3 Self-driving cars

Companies that build such types of driving, as well as self-driving cars such as Tesla, must teach a computer to master some essential parts of driving using systems digital sensors instead of the human mind.

To do this, companies usually start by training algorithms using a large amount of data.

You can imagine how a child learns through experiences constants and replication. These new services could provide models unexpected sales to companies.

2.2.4 Voice search and voice-activated assistants

One of the most common uses of DL is voice search and smart assistants voice-activated.

With the big tech giants have already made important Investments in this area, voice-activated assistants can be found on almost all smartphones.

Apple's Siri has been on the market since October 2011. Google today, the voice-activated assistant for Android, has been launched less one year after Siri. The newest voice-activated smart assistant is Amazon's Alexa.

2.2.5 machine translation

This is a task in which words, phrases or sentences given in one language are automatically translated into another language.
Machine translation has been around for a long time, but DL makes it possible to obtain the better results in two specific areas:

- Automatic text translation
- Automatic image translation

Text translation can be done without any prior processing of the sequence, which allows the algorithm to learn dependencies between words and their correspondence with a new language

2.2.6 Automatic text generation

This is an interesting task, where a corpus of text is learned and from this template a new text is generated, word by word or character by character.

The model is able to learn how to spell, punctuate, form sentences and even capture the style of the text in the corpus. The big Recurrent neural networks are used to learn the relationship between elements in the input string sequences, and then to generate text.

2.2.7 Image recognition

Another popular area when it comes to DL is the image recognition. Its purpose is to recognize and identify people and objects in images, as well as understanding the content and context.

The Image recognition is already used in several sectors such as games, social media, retail, tourism, etc. This task requires classification objects in a photo from a set of previously known objects. Complex of this task, called object detection, consists in specifically identifying one or more objects in the scene of the photo and draw a frame around them.

2.2.8 Automatic colorization

Colorizing the image poses the problem of adding colors with black and white photographs.

The DL can be used to use objects and their context in photography to color the image, much like a human operator could address the problem. This capability takes advantage of neural networks of high quality and very large convolution formed for ImageNet and co-opted for the problem of image colorization.

Generally, the approach involves the use of very large convolutional neural networks and layers supervised that recreate the image with the addition of colors.

2.2.9 detection of brain cancer

A team of French researchers noted that it was difficult to detect invasive brain cancer cells during a surgery, in part because of the effects of lighting in the rooms of operation. They found that using neural networks together with Raman spectroscopy during operations allowed them to detect Cancer cells more easily and reduce residual cancer after the operation.

2.2.10 Marketing Research

In addition to looking for new features DL can also be useful in the background. Market segmentation, marketing campaign analysis and many more can be improved using regression and DL classification models. That will aid in the case of possessing of large amounts of data[26]

2.3 Neural Networks

The practice of all DL algorithms are neural networks [22]. Neural Networks, also called ANN, are information processing models that simulate the functioning of a biological nervous system. It's similar to how the brain manipulates information on an operational level. All neural networks are consisting of interconnected neurons that are organized into layers (Figure 13) [22].

2.3.1 The neuron

What forms neural networks are artificial neurons inspired by the real neuron that exists in our brain. The following 2 figures show a representation of a real neuron and an artificial neuron:

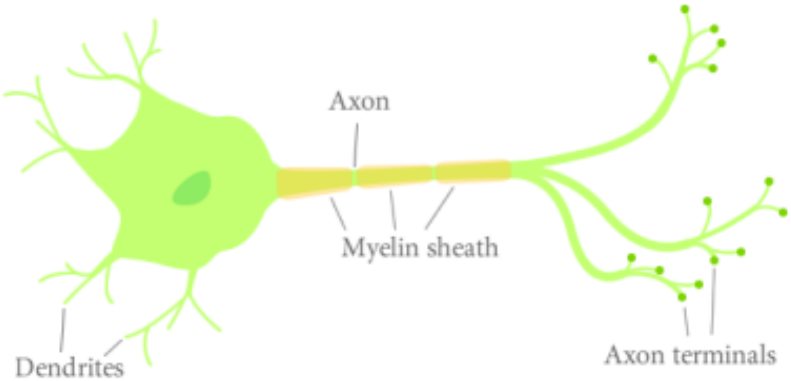


Figure 10: A real neuron

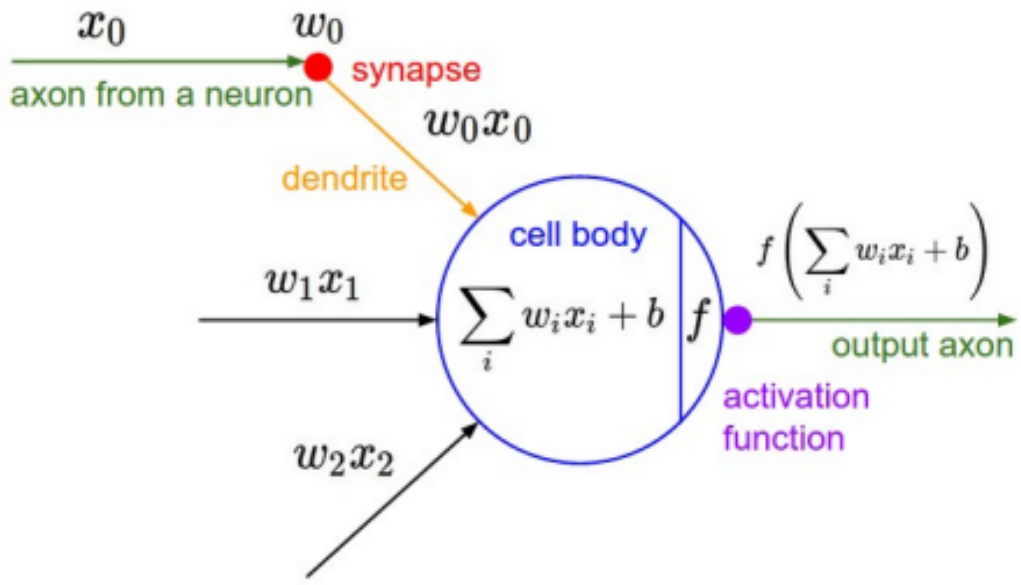


Figure 11: An artificial neuron

The Xi are numeric values that represent either the input data or the outputted values of other neurons. Wi weights are numerical values that represent either the input power value, which is the power value of connections between neurons. There are operations that take place at the level of the artificial neuron. The artificial neuron will do a product between the weight (W) and the input value (X), then add a bias (B), the result is passed to an activation function (F) that will add some non-linearity.

2.3.2 Activation Functions :

After the neuron has performed the product between its inputs and its weights, it also applies a non-linearity on this result. This Nonlinear function is called the activation function.

The activation function is an essential component of the neural network. What this Function has decided is whether the neuron is activated or not. It calculates the weighted sum of the enters and adds bias. It is a nonlinear transformation of the input value.

After the transformation, this output is sent to the next layer. Non-linearity is so important in neural networks, without the activation function, a network of neurons has become simply a linear model. There are many types of these functions, among which we find [27] [28]:

- **The Sigmoid Function** : this function is one of the most commonly used. It is bounded between 0 and 1, and it can be interpreted stochastically as the probability that the neuron activates, and it is usually called the logistic function or the logistics sigmoid.

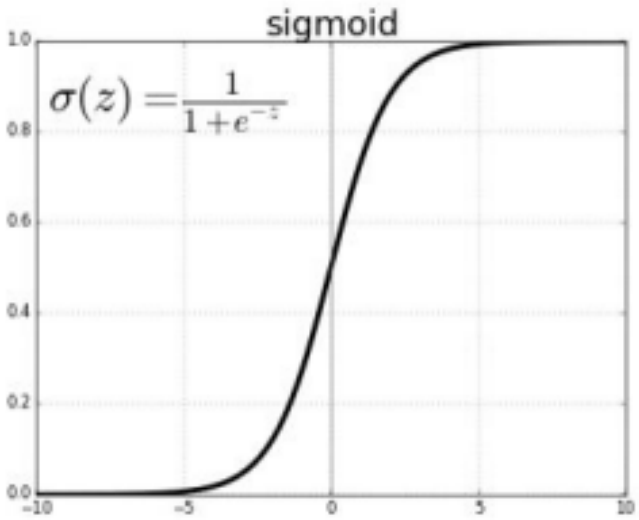


Figure 12: The Sigmoid function [27]

- **The ReLu Function** : the ReLu function is probably the closest to its biological correspondent [28]. This feature has recently become the choice of many tasks (especially in computer vision) [27]. As in the figure above, this function returns 0 if the input z is less than 0 and returns z itself if it is larger than 0.

- **The Softmax Function** : softmax regression (synonyms: Multinomial logistics,Maximum entropy classifier, or simply multi-class logistic regression) is a generalization of logistic regression that we can use for multi-class classification [29].

  Unlike other types of functions, the output of a neuron in a layer using the softamx function depends on the outputs of all other neurons in its layer. This is explained by the fact that it requires the sum ofall outputs to be equal of 1. In [27].

According to Hinton, in [31], we can deem a neural network is deep when the number of hidden layers surpasses 1. In this section, we will present the common structures of deep neural networks.

2.3.3 Fully connected networks:

A fully connected network allows you to transform an input list into an output list. The transformation is called fully connected because any input value can assign any output value. These layers will have many learnable parameters, even for relatively small inputs [30], but they differentiate by not assuming any structure of its inputs. We perform a series of transformation called calculations that alters the similarities between cases.

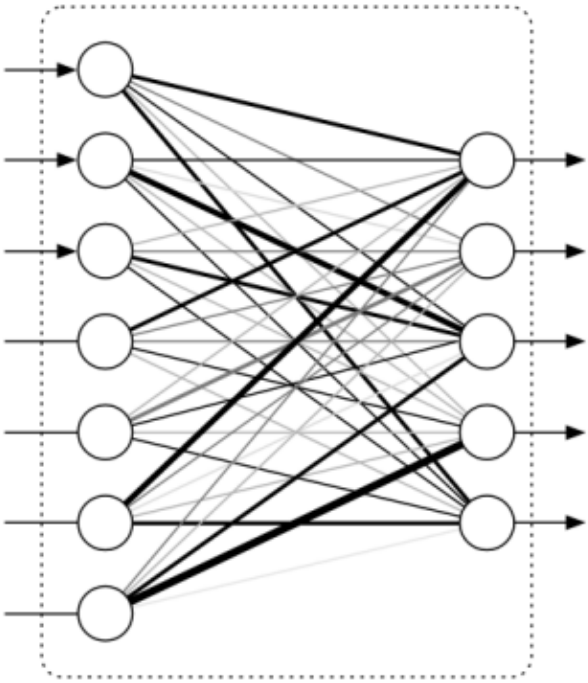


Figure 13: A fully connected network[30].

2.3.4 Convolutional networks:

A Convolutional Network (CNN: Convolutional Neural Networks) assumes a particular spatial structure in its input, In Particularly, it assumes that the entries that are close to each other in the input are semantically related [30]. CNN is a sequence of layers, and each layer transforms one activation volume into another by a differentiable function [33]. The three main types of layers for building this type of network are: convolutional layer, pooling layer and fully connected layer [33].

- **The convolutional layer:** It is the most important layer and the heart of the constitutive elements of the convolutional network, and it also performs the most heavy calculations.
- **The pooling layer:** It is common to periodically insert a Pooling layer into this type of architecture. Its function is to gradually reduce the spatial size of the representation to reduce the number of parameters and calculations in the network, and so also control overfitting.
- **The fully connected layer :** As mentioned earlier, neurons in a fully connected layer have full connections to all activation in the previous layer.

2.4 Residual Learning :

Before He et al[34] First introduced Residual Learning, the only way to improve the performance of a model is to increase the number of layers. However it was noticed that just stacking more layers will cause a performance degradation due to a couple of reasons we will mention on this section.

Yet, another problem risen with traditional Convolutional neural network was mentioned in [46] which is the harder optimization because when the model introduces more parameters, it becomes more difficult to train the network. This is not simply an overfitting problem, since sometimes adding more layers leads to even more training errors.

In fact the degradation problem occurs with/on the past CNNs when reaching the maximum threshold of depth. Figure 15 demonstrates the degradation problem (left), where it describes error rate on both training and testing data.

It is clearly seen that error is higher in the network with 56 layers than the one with 20 layers only. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error[34].

therefore deep CNNs, despite of having better classification performance, are harder to train.

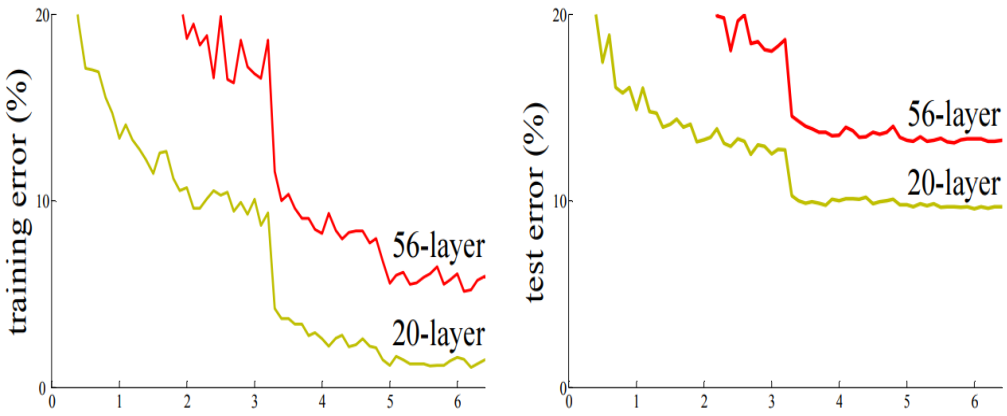


Figure 14: Training error (left) and test error (right) on a dataset with 20-layer and 56-layer[34].

To solve this problem, over the years many deep neural networks were being developed in order to make models more robust such as LeNet [35] which solved digit recognition problem by stacking up 7 layers one over the other.

After LeNet many variants of the CNN architecture were introduced like AlexNet [36], and GoogleNet [37], however accuracy degradation remains an observed problem due to all these deep networks having more number of convolution, pooling and activation layers stacked one over the other.

Before defining residual learning which solved this problem we have to look at the main concepts that led to the degradation problem and the concept that helped fixing it:

- **Cross Validation:** it is mainly how can we decide which machine learning method would be best for our dataset. Cross-validation is one of the most widely used data resampling methods to estimate the true prediction error of models and to tune model parameters [38].

- **Gradient Descent:** it's the most commonly used algorithm to optimize parameters of different machine learning models. Gradient Descent works efficiently when a given parameter can not be calculated using linear algebra calculations. So an optimization algorithm will take place in order to minimize the cost function. Since our goal is to minimize the cost function to find the optimized value for weights (parameters), we run multiple iterations with different weights and calculate the cost to arrive at a minimum cost as shown below[51].

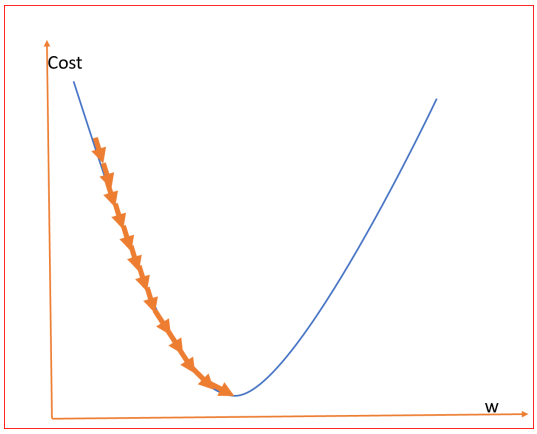


Figure 15: keep tracking the cost until we find the minimum (optimalcost)[51]

- **Back Propagation:** when training a Neural Network we pass our data into our model. The way this data flows through our model is via forward propagation where we're repeatedly calculating the weighted sum of the previous layers activation output with the corresponding weight.

  Then, passing the sum to the next layers activation function we do this until we reach the output layer, and at this point we calculate the loss on our output and then gradient descent will try to minimize this loss by first calculating the gradient of the loss function with respect to the weights, and then updating accordingly the weights in the network. To calculate this gradient, Gradient Descent uses backpropagation.

- **Vanishing Gradient:** it is the main reason of the degradation problem. This problem is a huge barrier to training deep neural networks. The backpropagation algorithm works by going from the output layer to the input layer, propagating the error gradient on the way.

Once the algorithm has computed the gradient of the cost function with regards to each parameter in the network, it uses these gradients to update each parameter with a Gradient Descent step. Unfortunately, gradients often get smaller and smaller as the algorithm progresses down to the lower layers. As a result, the Gradient Descent update leaves the lower layer connection weights virtually unchanged, and training never converges to a good solution[2]. The vanishing gradient problem in gradient-based training means the gradients of network weights approach zero. Therefore, it is difficult for the gradients to provide updates to the network weights[39]

So what is Residual Learning?

2.4.1 Definition :

Kaiming He et al. at [34]produced deep residual network (ResNet) which is almost like any other network that has convolution, pooling, activation, and fully-densed layers stacked one over the other.

He et Al.[34] defined residual learning as the process of adding an input x to the output (target function $H(x)$ of the network (i.e., you add a skip connection), then the network will be forced to mode $f(x) = h(x) - x$ rather than $h(x)$ [2].

So rather than expect stacked layers to approximate $H(x)$, we explicitly let these layers approximate a residual function $F(x) := H(x) - x$ The original function thus becomes $F(x) + x$

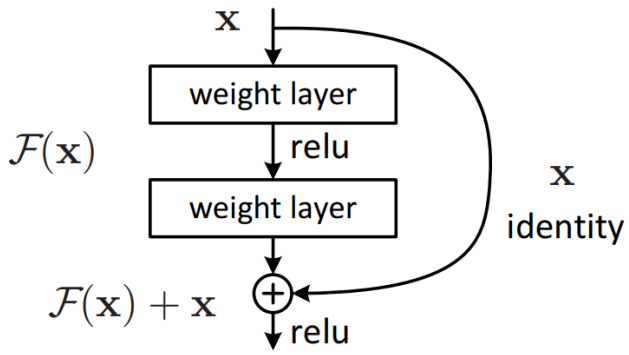


Figure 16: Residual learning: a building block[34].

In other words, without the skip connection which is considered as the core of residual blocks, our input simply will get multiplied by the weights of the layer then will be added a bias. Before being fed to the activation function to form out input which is $H(X) = f(X)$ But with the residual blocks the output will be different as shown in Figure 16 using the skip connection, the output is changed to $H(x) = f(x) + x$ Moreover, if you add many skip connections, the network can start making progress even if several layers have not started learning yet (See Figure 20). Thanks to skip connections, the signal can easily make its way across the whole network.

The deep residual network can be seen as a stack of residual units (blocks), where each residual unit is a small neural network with a skip connection [2].

In [34] they came across a problem that resulted from this skip connection, which is when the input and the output are not from the same dimensions, and in order to resolve it they settled on two approaches :

- The shortcut still performs identity mapping, with extra zero entries padded for increasing dimensions.
- The projection shortcut. is used to match dimensions (done by 1×1 convolutions).

2.4.2 Identity Function :

We can say that identity functions (identity mapping) are the optimal solution for the degradation problem traditional CNNs suffered.

ResNet came up with identity function which we can think of it as simple as adding the input of a layer to its output.

In other words, in order to approximate the final function of the block we add identity mapping to the input. In this way, if the identity mapping is already optimal and the stacked convolutional layers cannot learn more salient information, it can push the residual mapping to zero so as to avoid the degradation problem.

The residual module is shown to be easier to optimize so that deeper architecture can be developed[45].

In figure 17, we can clearly see that identity mapping liaise the input to the output

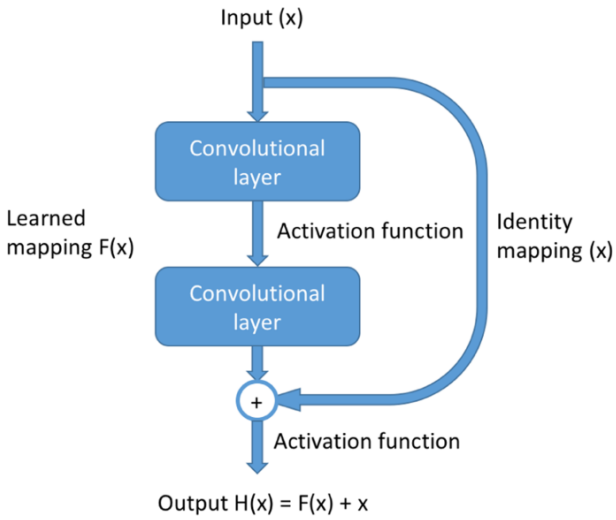


Figure 17: Identity Mapping[45].

2.4.3 How ResNet Helps ?

As we have discussed above, identity mapping is very helpful being the bridge between the input and the output of the layer.

Moreover identity mapping ensures that it won't be any degradation in performance, ResNet architecture when using identity mapping will surely guarantee that a given higher layer will either perform well or at least the same as the lower layer.

In other words, Say we have a shallow network and a deep network that maps an input x to output y by using the function $H(x)$. We want the deep network to perform at least as good as the shallow network and not degrade the performance as we saw in case of plain neural networks(without residual blocks).

One way of achieving so is if the additional layers in a deep network learn the identity function and thus their output equals inputs which do not allow them to degrade the performance even with extra layers[44].

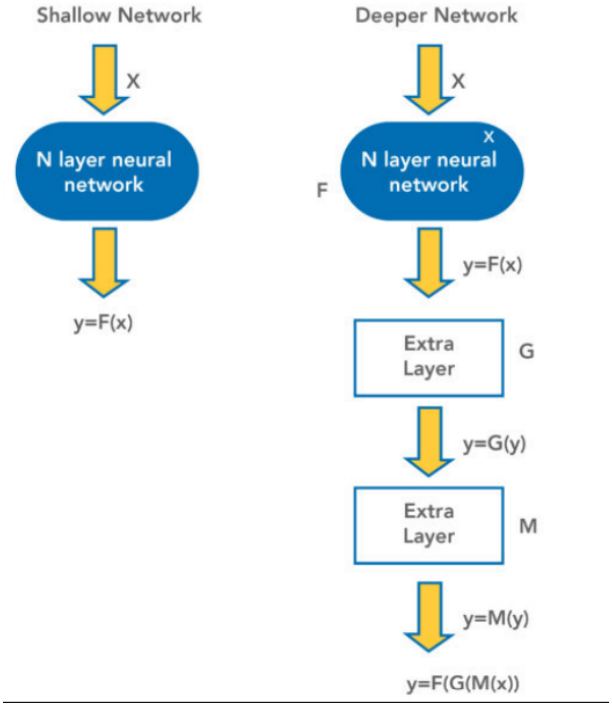


Figure 18: G and M act like an identity function on the deeper network.

It has been seen that residual blocks make it exceptionally easy for layers to learn identity functions. It is evident from the formulas above. In plain networks the output is $H(x) = f(x)$ So to learn an identity function, $f(x)$ must be equal to x which is grader to attain whereas incase of ResNet, which has output :

$$H(x)=f(x) + x$$

$$f(x)=0$$

$$H(x)=0$$

All we need is to make $f(x)$=0 which is easier and we will get x as output which is also our input, In the best-case scenario, additional layers of the deep neural network can better approximate the mapping of 'x' to output 'y' than it's the shallower counterpart and reduces the error by a significant margin.

And thus we expect ResNet to perform equally or better than the plain deep neural networks[44].
Figure 19 and table 4, illustrates the huge difference made by ResNet which leads us with three major observations :

- The situation is reversed with residual learning – the 34-layer ResNet is better than the 18-layer ResNet (by 2.8%). More importantly, the 34-layer ResNet exhibits considerably lower training error and is generalizable to the validation data. This indicates that the degradation problem is well addressed in this setting and we manage to obtain accuracy gains from increased depth[2].

- Compared to its plain counterpart, the 34-layer ResNet reduces the top-1 error by 3.5% (Table 4), resulting from the successfully reduced training error (Figure 19 right vs. left). This comparison verifies the effectiveness of residual learning on extremely deep systems[2].

- We also note that the 18-layer plain/residual nets are comparably accurate (Table 4), but the 18-layer ResNet converges faster (Figure 19 right vs. left).

When the net is "not overly deep" (18 layers here), the current SGD solver is still able to find good solutions to the plain net. In this case, the ResNet eases the optimization by providing faster convergence at the early stage[2].

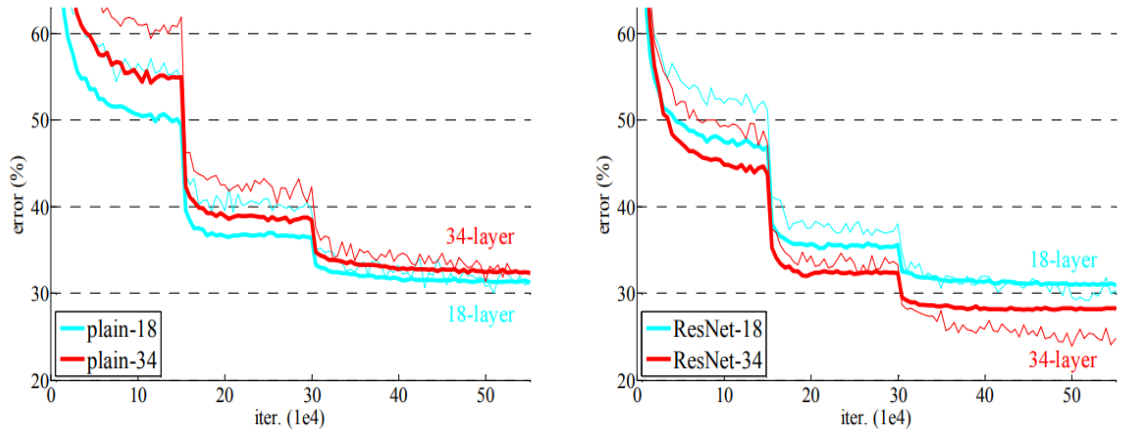


Figure 19: training error and validation error of plain network and resnet network[34].

	Plan	ResNet
18 layers	27.94	27.88
34 layers	28.54	25.03

Table 4: Table: ResNet has no more extra parameters in compared with the plain network. [34]

2.4.4 Residual Networks architecture:

ResNet network architecture in which they implemented the identity mapping, is inspired from the VGG network [43], no matter how deep the network is and its number of layers the structure remains the same. Which makes ResNet architecture special is the residual blocks it contains.

To better comprehend the structure as shown in Figure 20, overall the network receives an image as input with its height and width followed by a convolution block with its convolution layer with 7×7 kernel size and 16 filters and a max pooling layer with 3×3 kernel size. This was the initialization part of every ResNet architecture, it concludes with a global average pooling layer, a fully connected layer with 1000 neurons and a softmax.

In between the initialization and the end is just a very deep stack of simple residual units. Each residual unit is composed of two convolutional layers, with Batch Normalization and ReLU activation, using 3×3 kernels and preserving spatial dimensions (stride 1, padding is SAME)[2].

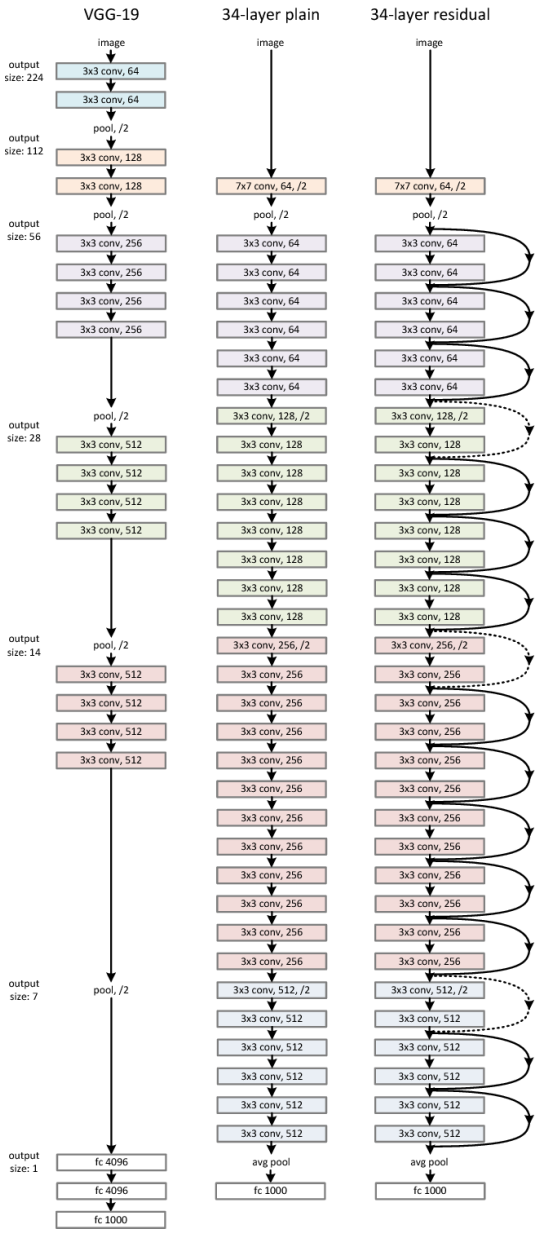


Figure 20: ResNet Architecture [34].

In figure 20 we notice a dashed connected and a curved arrow. The curved arrows refer to the identity connection. The dashed connected arrow represents that the convolution operation in the Residual Block is performed with stride 2, hence, the size of input will be reduced to half in terms of height and width but the channel width will be doubled, as we progress from one stage to another, the channel width is doubled and the size of the input is reduced to half[42].

Figure 21 illustrates more details and other variants of the structure of a 34 layer ResNet, with three residual blocks, followed by another four blocks and finally six residual blocks with a feature map dimensions of [64, 128, 256, 512] respectively.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 21: Detailed look on a 34 layers ResNet[34].

2.5 Conclusion:

In this chapter we have presented what is DL, its history, as well as neural networks. Then we have discussed the problem of the degradation which led to the Residual Learning which we explained in details. The next chapter we will see our contribution.

Chapitre 03

CONTRIBUTION

3 Contribution

3.1 Introduction:

In our quest to contribute into to the em-betterment of performance of IDS, we suggested a residual learning based model, and compared it with a CNN based one, inputs of our two models were the 10 attacks categories extracted from the UNSW-NB15 dataset[41]. We conducted several preprocessing experiments to validate our models. In what follows we describe the description of the dataset, different hyper-parameters that served us through our quest and then we will move on to the stage of experiments and implementations.

3.2 Tools:An Overview

3.2.1 Hardware:

Manufacturer	CPU	GPU	RAM
Lenovo IDEA-PAD 310	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz	Nvidua GeForce 920M	8 GB
HP 250 G6 Notebook	intel(R) Core i3-6006U CPU @ 2.00GHZ	intel(R) HD Graphics 520	4 GB

3.3 Software:

Software	Description
VS Code	code editore made by Microsoft. Optimal for building and debugging various applications
Python	Open source high level programming language designed to be easy to read and simple to implement
Google Colab	an executable document that lets write, run and share code within Google Drive enables us to execute Python code without any required setup on our own machine.

3.4 DataSet: our dataset contains 10 classes

Type	**No. Records**	**Description**
Normal	2,218,761	Natural transaction data.
Fuzzers	24,246	Attempting to cause a program or network suspended by feeding it the randomly generated data
Analysis	2,677	It contains different attacks of port scan ,spam and html files penetrations
Backdoors	1l—2,329	A technique in which a system security mechanism is bypassed stealthily to access a computer or its data
DoS	16,353	A malicious attempt to make a server or a network resource unavailable tousers, usually by temporarily interrupting or suspending the services of a host connected to the internet
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability
Generic	215,481	A technique works against all blockciphers (with a given block and key size), without consideration about the structure of the block-cipher
Reconnaissance	13,987	Contains all Strikes that can simulate attacks that gather information
Shellcode	1,511	A small piece of code used as the payload in the exploitation of software vulnerability
Worms	174	Attacker replicates itself in order to spread to other computers. ofter ,it uses a computer ntwork to spread itself, relying on security failures on the target computer to access it

Table 5: Table: UNSW-NB15 dataset distribution [41]

One look to the UNSW-NB15 dataset and you will come to visualize the class imbalance problem (Figure 22), considering the fact that normal class attacks constitute 87% of the dataset attacks leaving only 13% to the other nine types.

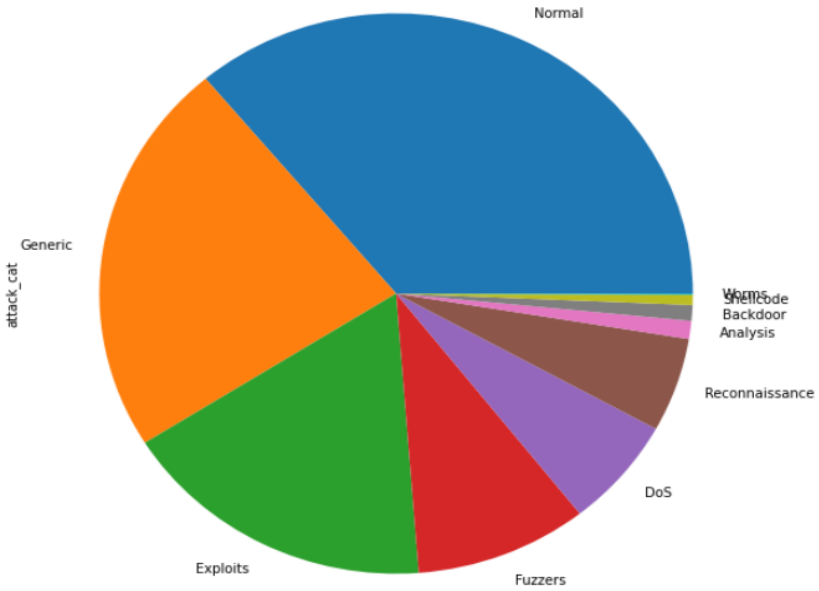


Figure 22: Distribution of UNSW-NB15 Dataset

3.5 Implementation:

3.5.1 DATA Preprocessing :

Data preprocessing is crucial in when dealing with deep learning models. Considering that UNSW-NB15 dataset has both numeric and symbolic features, data preprocessing provide some important steps that we need to adopt in order to ensure and enhance the performance.

- **Label Encoding :**

  Is it a very important step in datapreprocessing. In order to convert our labels to a machine-readable form we used label encoding, since UNSW-NB15 dataset as we mentioned earlier comes with both symbolic and numeric features. So it is very necessary that we convert our features into a numeric forms so that each feature of our 10 categories will have a unique integer value that will be outputed as an array.

- **Min-max Normalization :**

  In order to normalize our data we use min-max normalization. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

  Where $x*$in the normalized value, X is the original value before the min-max Normalization, Xmin is the minimum value and the Xmax is the maximum value.

3.6 Model's Architecture:

- **Convolutional layer:** As we mentioned earlier it is the main part of the Convolutional Neural Network, it is in this layer that we specify the number of filters (Kernels) which are the useful in such problems like classification. As well as we can consider numerous convolutional layers in a way that the output of a convolutional layer becomes the input of the next one.

- **Dropout :**In order to prevent overfitting we apply dropout which is the most used regularization method.

  Regularization is any modification we make to a learning algorithm with the aim of reducing its generalization error but not its learning error[47].

  We mainly use the dropout method to the output of some network layer, as shown in figure 23 it will randomly and periodically remove some of the neurons as well as their input and output connections.

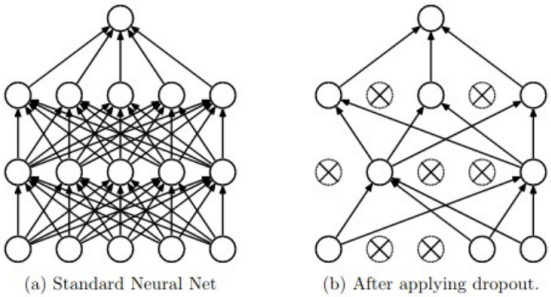


Figure 23: Standard neural network and after applying dropout [47]

When the Dropout is executed as show in figure 21 and some neurons are removed during training, the other neurons got to respond and fit the representation required to make predictions for the missing neurons, this method improves generalization because it forces the layers to learn the same concept with different neurons.

3.6.1 Focal Loss:

As mentioned before, the UNSW-NB15 we are using is imbalanced. In order to address said problem our model uses focal loss function. Focal loss was proposed in [48]to balance the loss between dataset's features and enhance the detecting capabilities .

it helps the network focus on hard classified objects, in case they are overwhelmed by a large number of easily classified objects[49]. in other words focal loss function was applied on our proposed model for a better detection of minor classes. Focal loss enabled the model to concentrate on features that are harder to learn, and indeed the acquired results indicate that the focal loss apprehended complex features adequately.

3.7 Models :

During our experiments, we created two models with different architectures, where the first was a CNN (Convolutional Neural Networks) model and the second was a ResNet (Residual Network) model. In the following, we present the architecture of these two:

The following figure represents the different layers contained in our model :

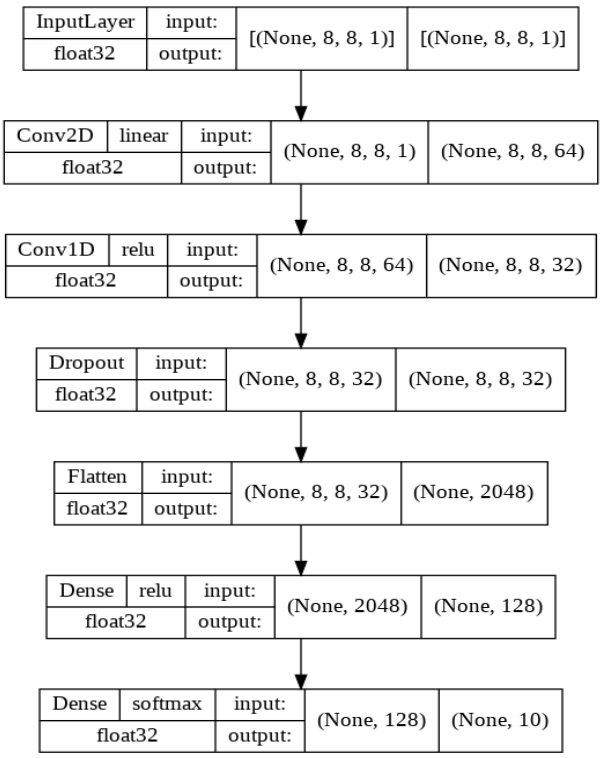


Figure 24: CNN model's architecture.

- **Conv2D layers:** we used two-dimensional convolutional layers.
- **Dropout Layer:**we have applied one Dropout layer for the sake of regularization.
- **Flatten Layer:**we add a flatten layer.

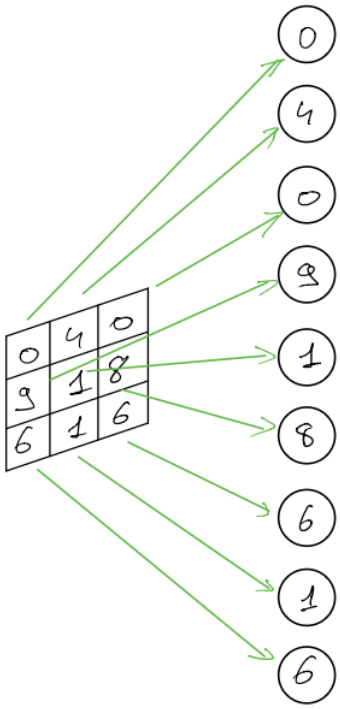


Figure 25: The role of the Flatten Layer[50]

- **A fully connected layer:** we used two fully connected layers with 128 hidden neurons and a softmax activation function to predict the output.

3.8 Results and discussions:

- ResNet Model:

layer name	output size	18-layer	34-laye	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, strid 2				
		[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 26: ResNet Model

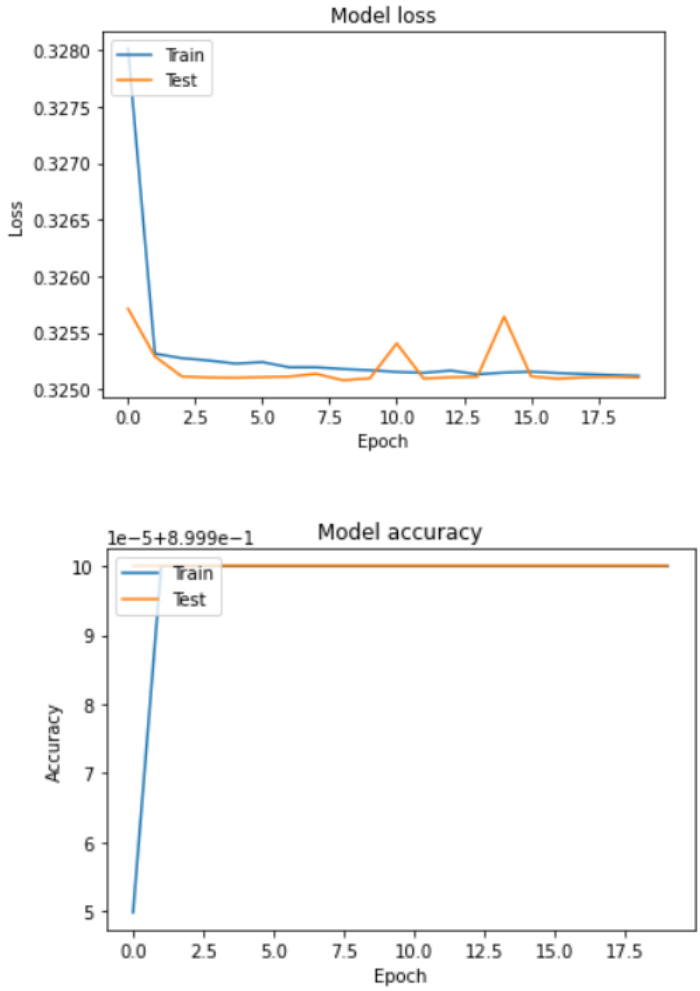
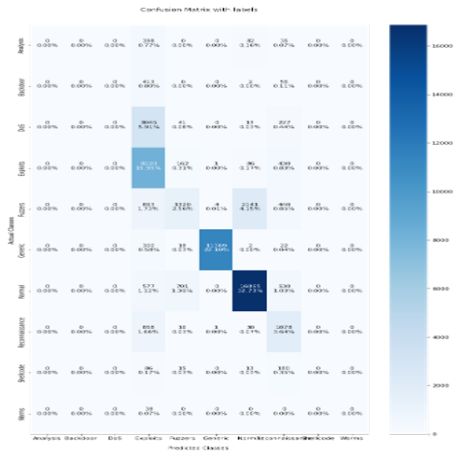


Figure 27: Loss and accuracy of the CNN model.



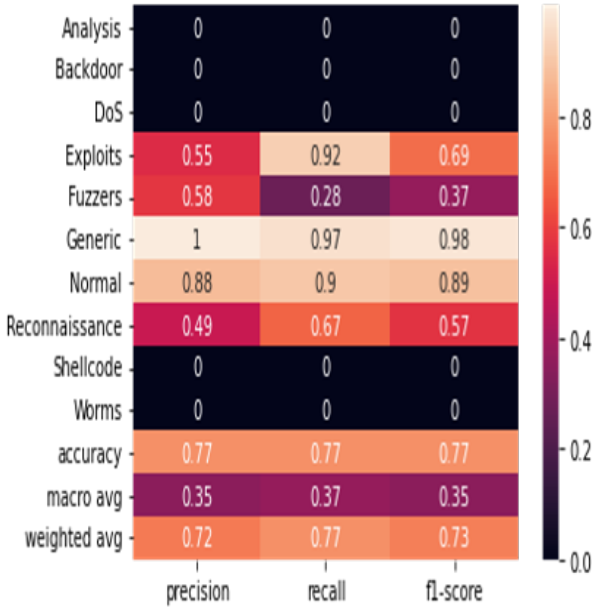


Figure 28: Confusion matrix and Classification Report of CNN model.

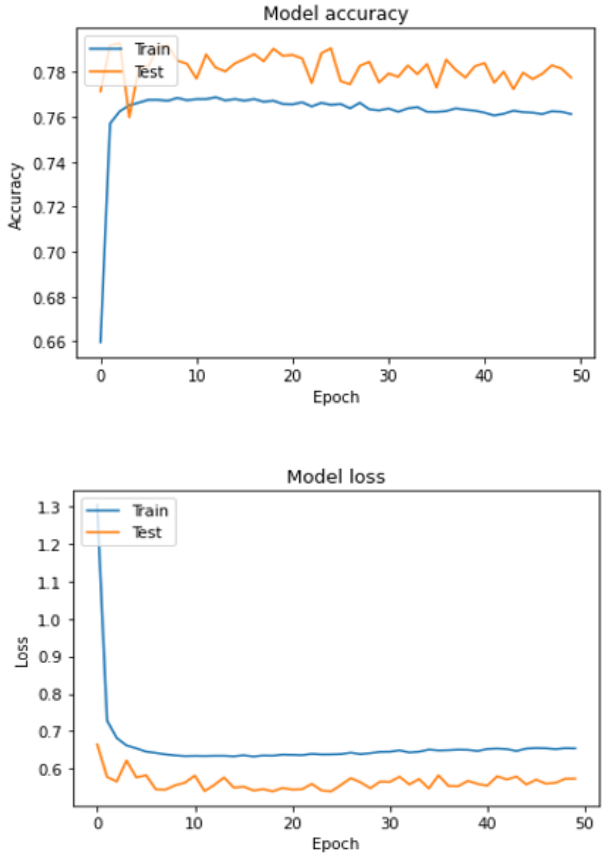


Figure 29: Accuracy and loss rates of the ResNet model

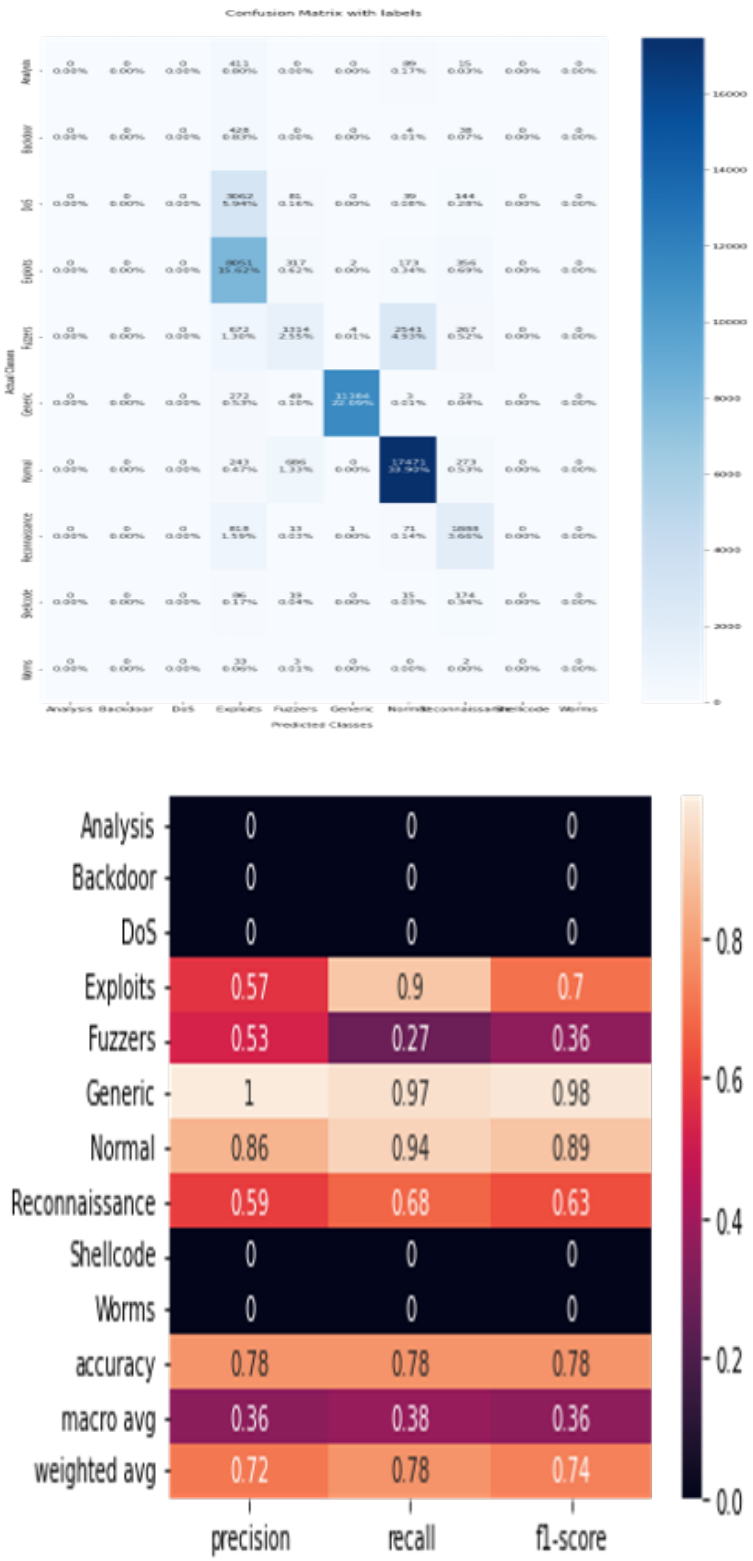


Figure 30: Confusion matrix and Classification Report of the ResNet model

As we observe in figures 27, 28, 29, 30 the Residual network model with 50 layers(ResNet50), outperformed the CNN model, achieving higher accuracy rates and lower loss values, shows us the advantage of using residual networks in network intrusion detection field.

3.9 Conclusion:

In this chapter we have seen our share of contribution to the problem of Intrusion Detection, the experiments we have made with the different parameters allow us to improve the performance of our model in terms of accuracy and error rate. Representing the tools and the dataset we used, as well as the improvements done due to our adjustments, to obtain better results and furthermore to make the comparison for two different models.

General Conclusion

In this work, we sought out to explore the vast field of intrusion detection which, like all other areas of cyber security, has been crossbred with Machine Learning and deep learning thus resulting in a major evolution as depicted in this project.

Prior to our engagement, we invested a lot of time consulting and reviewing the proper scientific documentation to gain an appropriate perspective on how to apply a Residual Learning model to our problem.

this work enabled us to utilize our knowledge of Deep Learning and to further develop them , most importantly it led us to the discovery of residual networks which brings out deep learning's best potential.

We consider this work a gateway towards expanding our horizon, we hope that in the coming endeavors we shall take on more complicated intrusion detection datasets and approaching it in two different ways :

- Building our own ResNet model.
- Utilizing Transfer Learning which will serve in the development of our research results.

Which we believe that it will achieve outstanding results toward the intrusion detection field.

References

[1] INTRODUCTION TO COMPUTATIONAL THINKING AND DATA SCIENCE – MIT – OCW.

[2] Hands-On Machine Learning with Scikit-Learn - TensorFlow, Aurélien Géron, 2017

[3] Python Machine Learning, Sebastian Raschka, 2015

[4] A study on classification techniques in data mining. in Computing, Communications and Networking Technologies (ICCCNT). Kesavaraj G, Sukumaran S. 2013 Fourth International Conference on, 2013; pp. 1-7.

[5] Mining educational data to analyze students' performance. arXiv preprint arXiv:1201.3417. Baradwaj BK, Pal S, 2012.

[6] "Personalized Collaborative Filtering Recommendation Algorithm based on Linear Regression," in 2019 IEEE International Conference on Power Data Science (ICPDS). J. Wu, C. Liu, W. Cui, and Y. Zhang, 2019, pp. 139-142

[7] Support Vector Machines for Classification, Mariette Awad Rahul Khanna, 2015

[8] An Introduction to Statistical Learning with Applications in R, Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. 2021.

[9] Semi-supervised learning: a brief review, Viswanath Pulabaigari, Eswara Reddy B. 2018

[10] MACHINE LEARNING: An Algorithmic Perspective, Stephen Marsland. 2015

[11] Intrusion Detection Algorithm Based on Convolutional Neural Network, Yuchen Liu, Shengli Liu. 2017

[12]] An Introduction to Intrusion-Detection Systems, Hervé Debar. 2009.

[13] Network Intrusion Detection and its strategic importance, Muhammad Kamran Asif, Talha ALI Khan. 2013.

[14] Artificial intelligence in network intrusion detection, Miroslav Stampar, 2015

[15] A Review on Hybrid Intrusion Detection System using Artificial Immune System Approaches, Nidhi Chandra, Pavitra Chauhan, 2013

[16] Live traffic analysis of tcp /ip gateways. Proc. ISOC Symposium on Network and Distributed System Security (NDSS98), A. Phillip, Porras and Alfonso Valdes. (San Diego, CA, March 98), Internet Society. 1998.

[17] A revised taxonomy for intrusion detection systems. Annales des

[18] Tarek Abbes. (2004) Classification du trafic et optimisation des règles de filtrage pour la détection d'intrusions . Tarek Abbes. Thèse de doctorat de l'université Henri Poincaré.Nancy1 .2004

[19] ImageNet classification with deep convolutional neural networks. Krizhevsky, A.Sutskever, I. Hinton. Advances in Neural Information Processing Systems. 2012.

[20] Deep Neural Networks for Acoustic Modeling in Speech Recognition. Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury.

[21] Deep Learning Yann LeCun, Yoshua Bengio Geoffrey Hinton. 2015

[22] Python Deep Learning. Ivan Vasilev, Daniel Slater, Gianmario Spacagna, Peter Roelants, Valentino Zocca. 2019

[23] https ://ai.google, 20 Avril 2019

[24] Java Deep Learning Essentials. Yusuke Sugomori. 2016

[25] Deep Learning with Keras. Antonio Gulli, Sujit Pal. 2017

[26] Top 15 Deep Learning applications that will rule the world in 2018 and beyond, Vartul Mittal, 3 Oct 2017

[27] Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms. Nikhil Buduma. 2017

[28] Python Deep Learning. Valentino Zocca, Gianmario Spacagna, Daniel Slater, Peter Roelants. 2017.

[29] https ://sebastianraschka.com/faq/docs/softmax regression.html, 20 Mars 2019.

[30] TensorFlow for Deep Learning. Reza Bosagh Zadeh, Bharath Ramsundar. 2017

[31] Deep Learning with Keras. Antonio Gulli, Sujit Pal. 2017

[32] https ://www.cs.toronto.edu/tijmen/csc321/slides/lecture slides lec2.pdf . 2014

[33] http ://cs231n.github.io/convolutional-networks/overview 25 Mars 2019.

[34] Deep Residual Learning for Image Recognition, He et al. 2015.

[35] Gradient-Based Learning Applied To Document Recognition, LeCun et al. , 1998

[36] ImageNet Classification with Deep Convolutional Neural Networks, Krizhevsky et al. , 2012

[37] Going Deeper with Convolutions, Szegedy et al. , 2014

[38] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2nd edition, Springer, NewYork/Berlin/Heidelberg, 2008.

[39] Handling Vanishing Gradient Problem Using Artificial Derivative, Hu et al. ,2017

[40] Introduction to Resnet or Residual Network, Hussain Mujtaba, 2020

[41] UNSW-NB15: a comprehensive data set for network intrusion detection systems, Moustafa et al. 2015

[42] Detailed Guide to Understand and Implement ResNets, Ankit Sachan. 2019

[43] Very deep convolutional networks for large-scale image recognition. In ICLR, K. Simonyan and A. Zisserman. 2015

[44] Introduction to ResNet or Residual Network, Hussain Mujtaba. 2020

[45] Learning to Predict Crystal Plasticity at the Nanoscale: Deep Residual Networks and Size Effects in Uniaxial Compression Discrete Dislocation Simulations, Yang et al. 2015

[46] Highway networks, R. K. Srivastava, K. Greff, and J. Schmidhuber. 2015.

[47] Deep Learning. Ian Goodfellow, Yoshua Bengio, Aaron Courville. MIT Press. 2016.

[48] Focal loss for dense object detection. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. 2020.

[49] Focal Loss in 3D Object Detection, Yun et al. 2019.

[50] https://erwanscornet.github.io/teaching/CNN.pdf.

[51] Machine learning : Gradient Descent, Renu Khandelwal2018.

ملخص

على مر السنين، سعى الباحثون إلى تحسين أداء أنظمة اكتشاف اختراق الشبكة.. أثبتت العديد من حلول التعلم الألى والتعلم العميق فعالياتها في اكتشاف اختراق الشبكة وفي مساهماتنا اخترنا انشاء نموذج هجين بين التعلم المتبقي وأنظمة اكتشاف اختراق الشبكة. النموذج المتحصل عليه لديه دقة أفضل مقارنة بالنماذج الحالية

الكلمات المفتاحية : تعلم الالة، التعلم العميق، نظام اكتشاف اختراق الشبكة، التعلم المتبقي.

Abstract

Over the years, within the academic circles, researchers sought out to enhance the performance of Intrusion Detection Systems. Many Machine Learning and Deep Learning solutions proved efficient in network intrusion detection and in our contribution, we opted to create a cross-breed model between Residual Learning and network Intrusion Detection. The resulted model has a better accuracy compared to existing models.

Key words : machine learning, deep learning, intrusion detection system, residual learning.

Résumé

Au cours des années, au sein des cercles académiques, les chercheurs ont visés à améliorer les performances des systèmes de détection d'intrusion. De nombreuses solutions de l'apprentissage machine et l'apprentissage profond ont démontrées leurs efficacités dans la détection d'intrusion réseau, dans notre contribution nous avons opté pour créer un modèle de détection d'intrusion réseau basé sur l'apprentissage résiduel . Le modèle obtenu a une meilleure précision par rapport aux modèles existants.

Mot clés : apprentissage machine, apprentissage profond, system de détection d'intrusion, apprentissage résiduel