الجمـهورية الجزائرية الديمقراطية الشعبيـة
وزارة التعليم العالي والبحث العلمي
جامعة سعيدة د.مولاي الطاهر
كلية التكنولوجيا
قسم: الإعلام الآلي

Master Thesis

Specialty: Computer Security and Cryptography

Theme

Comparing Data Presentation Techniques for CNN-based IDS

Presented by :

Lamia Mekhalfi

Ikram Zerrouki

Supervised by:

Dr.Hassene Chaibi

University Year 2022-2023

Abstract

This study aimed to investigate the effectiveness of different data presentation techniques in the context of CNN-based Intrusion Detection Systems (IDS). The rapid evolution of technology and associated risks have highlighted the inadequacy of traditional security measures in safeguarding files, data, and personal information. To address this, IDS/IPS solutions have been employed to secure networks and protect against vulnerabilities. However, the emergence of deep learning methods, including Convolutional Neural Networks (CNNs), has provided new possibilities for addressing cybersecurity challenges.

This study focused on techniques for presenting non-image data by transforming it into visual representations for exploiting the inherent image recognition capabilities of CNNs. By exploring various data presentation techniques, the research aimed to improve the performance of IDS. Two distinct datasets were utilized for experimentation: the NSL-KDD dataset, widely employed as a benchmark in intrusion detection research, and the CICIDS2017 dataset.

Through comparative analysis, the study compared the results obtained from different image frames and CNN-based classification techniques with Random Forest, aiming to determine the most effective data presentation technique for CNN-based IDS. The findings of this research provide valuable insights into optimizing CNN-based IDS approaches, ensuring precision and short execution time.

Overall, this study contributes to the field of cybersecurity by shedding light on the role of data presentation techniques in CNN-based IDS. By understanding the strengths and weaknesses of various approaches, organizations can enhance their network security measures and make informed decisions regarding the deployment of IDS solutions.

Keywords: CNN, Data presentation, IDS/IPS, NSLKDD,CICIDS2017, Random Forest .

Résumé

Ce mémoire visait à étudier l'efficacité de différentes techniques de présentation des données dans le contexte des systèmes de détection d'intrusions (IDS) basés sur les réseaux de neurones convolutionnels (CNN). L'évolution rapide de la technologie et les risques associés ont souligné l'insuffisance des mesures de sécurité traditionnelles pour protéger les fichiers, les données et les informations personnelles. Pour remédier à cela, des solutions IDS/IPS ont été mises en place pour sécuriser les réseaux et se prémunir contre les vulnérabilités. Cependant, l'émergence de méthodes d'apprentissage en profondeur, notamment les CNN, offre de nouvelles possibilités pour relever les défis de la cyber sécurité.

Cette étude s'est concentrée sur les techniques de présentation des données non graphiques en les transformant en représentations visuelles pour exploiter les capacités inhérentes de reconnaissance d'images des CNN. En explorant différentes techniques de présentation des données, la recherche visait à améliorer les performances des IDS. Deux ensembles de données distincts ont été utilisés pour les expérimentations : le jeu de données NSL-KDD, largement utilisé comme référence dans la recherche sur la détection d'intrusions, et le jeu de données CICIDS2017.

Par le biais d'une analyse comparative, l'étude a comparé les résultats obtenus à partir de différentes structures d'images et de techniques de classification basées sur les CNN avec Random Forest, dans le but de déterminer la technique de présentation des données la plus efficace pour les IDS basés sur les CNN. Les conclusions de cette recherche fournissent des connaissances précieuses pour optimiser les approches IDS basées sur les CNN, en garantissant une précision et un temps d'exécution courts.

Dans l'ensemble, cette étude contribue au domaine de la cyber sécurité en mettant en lumière le rôle des techniques de présentation des données dans les IDS basés sur les CNN. En comprenant les forces et les faiblesses des différentes approches, les organisations peuvent améliorer leurs mesures de sécurité réseau et prendre des décisions éclairées quant au déploiement de solutions IDS.

Mots-clés: CNN, Data presentation, IDS/IPS, NSLKDD, CICIDS2017, Random Forest .

ملخص

كان الهدف من هذه الدراسة هو اختبار فعالية تقنيات تصور البيانات المختلفة في سياق أنظمة كشف التسلل القائمة على الشبكة العصبية التلافيفية، أظهر التطور السريع للتكنولوجيا والمخاطر المرتبطة بها أن التدابير الأمنية التقليدية لحماية الملفات والبيانات والمعلومات الشخصية غير كافية. لحل هذه المشكلة ، تم تنفيذ حلول نظام كشف التسلل / نظام منع التسلل لتأمين الشبكة والحماية من نقاط الضعف. ومع ذلك ، فإن ظهور أساليب التعلم العميق ، بما في ذلك الشبكات العصبية التلافيفية، قد أوجد طرقا جديدة لمواجهة تحديات الأمن السيبراني. ركزت هذه الدراسة على تقنيات تمثيل البيانات غير المصورة عن طريق تحويلها إلى تمثيلات مرئية لاستغلال قدرات التعرف على الصور الكامنة في الشبكات العصبية التلافيفية. من خلال فحص تقنيات عرض البيانات المختلفة ، يهدف البحث إلى تحسين أداء نظام كشف التسلل. تم استخدام مجموعتي بيانات مختلفتين للتجارب: مجموعة بيانات NSL-KDD ، والتي غالبا ما تستخدم كمعيار في أبحاث كشف التسلل ، ومجموعة بيانات CICIDS2017. من خلال تحليل مقارن ، قارنت الدراسة نتائج الصور المختلفة وتقنيات التصنيف المستندة إلى الشبكة العصبية التلافيفية مع Random Forest ، بهدف تحديد تقنية عرض البيانات الأكثر لنظام كشف التسلل القائم على الشبكة العصبية التلافيفية. توفر نتائج هذه الدراسات معلومات قيمة لتحسين أساليب نظام كشف التسلل القائم على الشبكة العصبية التلافيفية ، مما يضمن الدقة ووقت التنفيذ القصير. بشكل عام ، تساهم هذه الدراسة في مجال أمن المعلومات من خلال تسليط الضوء على دور تقنيات عرض البيانات في أنظمة نظام كشف التسلل القائم على الشبكة العصبية التلافيفية. من خلال فهم نقاط القوة والضعف في الأساليب المختلفة ، يمكن للمؤسسات تحسين تدابير أمان الشبكة واتخاذ قرارات مستنيرة بشأن تنفيذ حلول نظام كشف التسلل.

الكلمات المفتاحية: الشبكة العصبية التلافيفية، عرض البيانات، نظام كشف التسلل / نظام منع التسلل، NSLKDD. CICIDS2017 ، Random Forest.

.

Acknowledgements

First and foremost, we are grateful to Allah Almighty for providing us the perseverance and strength to complete our search.

This research is a loving tribute to our parents, who have been there for us financially, emotionally, and spiritually. To our siblings, cousins, friends, professors over through years , and family.

We would like to extend our appreciation to our supervisor**, Dr. Hassene Chaibi**, for all of the guidance, help, inspiration, and support. We are really grateful for his tolerance with our inquiries and ignore to some points.

Additionally, we would want to sincerely thank ourselves for our tremendous patience and work.

Dedicates

This study is dedicated to everyone who illuminated the mind of another with his knowledge or delirious with the correct answer.

As well to my first role model, and my beacon that illuminates my path, to the ones who gave me and continues to give me without limits, to the ones who raised my head high in pride of them **my dear father and beloved mother** (may God keep them as an asset for me) the shadow that I shelter in at all times to the candles that illuminate the way for me, to my sister **Ferdous,** who never stopped inspiring me with either her powerful words or her brilliant thoughts. A great appreciation to all the members of my family and friends as well.

Special thanks also to my dear friend **Lamia** for her company and efforts throughout this thesis or for her friendship.

Ikram

First things first, I thank **Allah Almighty** for everything and I dedicate my *work* to **my grand father** who had the credit for raising me and bringing me to this point (may God rest his soul),and a special feeling of gratitude to **my loving parents** for their support, encouragement and for providing me financially, thanks to my sister '**Lila**' and my brother '**Mohamed**' and all **my family** and **my friends** who made dua'a for me.

Then I would like to thank my best friend **Afraa** for everything her encouragement, her support and her work.

Last but not least I want to thank me for believing in me .I want to thank me for all the hard work that I did especially in the last 5 years and I want to thank me for my great patience and effort.

Lamia

And a heartfelt thanks to our wonderful instructor, the jury, and everyone who will read this. We also wish to express our sincere love and appreciation to all of our loved ones who were unable to share in this moment with us and for us (may God rest their souls).

Acronyms List

IDS	Intrusion Detection System
AI	Artificial Intelligence
CNN	Convolutional Neural Network
ML	Machine Learning
IT	Information Technology
IETF	Internet Engineering Task Force
IoT	Internet of Things
DL	Deep Learning
SQL	Structured Query Language
LSTM	Short Term Memory Units
GRU	Gated Recurrent Unit
PCA	Principal Component Analysis
DoS	Deni of Service
Probe	Probing
U2R	User to Root
R2L	Remote to Local
DDoS	Distributed Denial of Service
PCAP	PaCket CaPture
IP	Internet Protocol
PCA	Principal Component Analysis
CSV	Candidate Support Vector
HTTP	Hyper-Text Transform Protocol
HTTPS	Hyper-Text Transform Protocol Security
ACC	Accuracy
PVV	Precision and positive predictive value
TRP	True Positive Rate

Acronyms List

FNR	False Negative Rate
TNR	True Negative Rate
FPR	False Positive Rate
AUC	Area Under the Curve
RNN	Recurrent neural network
ENV	Environment
COVID-19	Coronavirus Disease 19
ReLu	Rectified Linear Unit

Contents

Contents

Tables List

Figures List

Figures List

General Introduction

General Introduction

Introduction

Information and communication technologies have advanced along with networks, particularly Internet networks, and now provide us with necessary services for distance learning, online shopping and paying, instant messaging, videoconferencing, and many other emerging technologies like vending machines and autonomous vehicles, among others. The widespread use of these computer tools has, however, led to the development of new security flaws.

Nowadays, networks and interconnected information systems have encountered real malicious or unintentional risks. There doesn't seem to be a day that goes by where we don't see an announcement of a new story about cyber-security issues, whether it be privacy hacking through social media, credit card fraud, economic espionage, computer system infection is criticized by attacks denial of service, or many other cyber threats pose problems and challenges major in the coming years. In the midst of the Covid-19 issue, Paris hospitals were recently targeted on March 23, 2020, by a cyber-denial of service assault. 19 (coronavirus). Today, these cyber-security dangers are one of the top worries and the most popular sector for IT (Information Technology) investment.

Context

This thesis is set in a context where the rapid evolution of technology and increasing security threats has highlighted the need to rethink data protection approaches. Traditional security methods have proven inadequate in addressing sophisticated cyber-attacks that compromise the confidentiality and integrity of files, data, and personal information. Convolutional Neural Network (CNN) based Intrusion Detection Systems (IDS) have emerged as a promising solution in the field of cyber security. By harnessing the image recognition capabilities of CNNs, it becomes possible to identify intrusion patterns and detect suspicious activities within networks. Therefore, this thesis is situated in a context where the application of deep learning techniques, such as CNNs, offers new possibilities to enhance network security and improve intrusion detection.

Problematic

The problem statement of this thesis revolves around the need to explore different data presentation techniques for CNN-based Intrusion Detection Systems (IDS). As technology rapidly evolves and security risks increase, it has become evident that traditional security measures are no longer sufficient to safeguard files, data, and personal information. IDS/IPS solutions have been developed to secure networks and mitigate vulnerabilities, but the emergence of deep learning methods, such as CNNs, offers new possibilities for addressing cyber security challenges. The central problem of this thesis lies in comparing various data presentation approaches, including transforming non-graphical data into visual representations, to maximize the performance of CNN-based IDS. The objective is to identify the most effective technique for presenting data to CNNs, thereby improving the precision and execution time of intrusion detection systems.

General Introduction

The purpose of work

This study aims to investigate different data presentation techniques, including the transformation of non-image data into visual representations, to leverage the powerful image recognition capabilities of CNNs. By comparing and analyzing the outcomes of various data presentation approaches, the research aims to identify the most effective technique for optimizing CNN-based IDS, ensuring enhanced precision, efficiency, and reliability in intrusion detection. The findings of this study will contribute to the field of cyber security by providing valuable insights into the optimization of CNN-based IDS approaches.

Our thesis is structured as follows:

- The first chapter provides an introduction to intrusion detection systems, including their operational principles and other relevant topics related to intrusion detection.
- The second chapter focuses on deep learning (DL) for intrusion detection, exploring various DL methods and their underlying principles of operation.
- The third chapter presents a comprehensive review of research on CNN-based intrusion detection systems within the context of deep learning.
- The fourth chapter outlines the established deep learning techniques and discusses the evaluation criteria employed to assess their effectiveness in intrusion detection. Additionally, this chapter presents the findings obtained from the evaluation process.

Chapter 1:
Intrusion Detection System

Chapter1: Intrusion Detection System

1.1Introduction:

Due to the risky system usage throughout the years, web and computer frameworks have produced a number of security concerns. According to CERT insights (Computer Emergency Response Team), the total number of disruptions has grown too much over time. Any malicious interruption or assault on computer systems, data storage systems, or other organizational weaknesses might result in real disasters and harm computer security measures isuch as Confidentiality, Integrity, and Availability (CIA). The risks to information security and arrangement are still major research concerns as of right now. IDS (intrusion detection systems) and its scientific classification have been covered in a variety of literary works.

1.2Defintion:

An Intrusion Detection System (IDS) may be a framework that screens network traffic for screens suspicious movement and issues alerts when such movement is found. It is a software application that scans a network or a system for the harmful activity or policy breaching. Any malicious wander or infringement is detailed either to an chairman or collected centrally employing a security data and event administration (SIEM) framework. A SIEM framework coordinating yields from numerous sources and employments alert sifting methods to distinguish malevolent activity from false alarms. In spite of the fact that interruption discovery frameworks screen systems for possibly noxious action, they are too inclined to untrue cautions. Organizations have to be fine-tune their IDS items to recognize what ordinary activity on the organize looks like as compared to pernicious action. Intrusion prevention systems too screen organize bundles inbound the framework to check the pernicious exercises included in it and at once send the caution notify.

1.3The basic model of an intrusion detection system:

The IETF has proposed the structure of an intrusion detection system, which consists of a few devices, each device having its own assignment. This system is designed to detect network intrusions at the first moment and enlighten the administrator or IT personnel on the plausibility of a network intrusion. The IETF's IDWG (Interruption Discovery Working Gather) includes and standardizes the structure of this system. Different IDS may combine these components into a single module, or have different occasions of these modules

- **The administrator** is responsible for establishing the security policy of the organization that deploys and configures IDS and supports the predefined declaration of activities to meet the needs of a system of information. [1]
- **The data Source** many types of information come from various sources in different ways (network, system, application and alarms). The IDS System has no constraints on the information sources used; it employs appropriate sensors to evaluate data from various sources and detect approved or unwanted unlawful behaviors. [1]
- Key elements of the system are **the sensor and the analyzer**. At first, the sensor will access the raw data, gather all the details of the actions taking place, and pass them as events to the analyst (sequence of activities). The latter will then examine these occurrences and report any unapproved or undesirable actions or occurrences that

could be of interest to the security administrator. The sensor and the analyzer are typically included in the same component of existing IDSs. [1]

- **The manager** is a crucial element that gives the operator the ability to control the system's numerous parts. Sensor configuration, analyzer configuration, event notification management, data aggregation, and report management are examples of common manager tasks, although they are not restricted to these. [1]
- Actions made in reaction to an incident are referred to as a **response**. It can be started by a person or automatically executed by an entity in the IDS architecture. It is a pretty typical response to notify the operator. Additional reactions can be activity monitoring, raw data logging (from the data source that described the incident), network or user shutdown or application session, or adjusting network or system access rules, among others. [1]

1.4Classification of Intrusion detection system:

- **Network Intrusion Detection System (NIDS):**

Network intrusion detection systems (NIDS) are set up at a predetermined location inside the network to monitor traffic from all connected devices. It carries out an observation of all subnet traffic passing through and compares that traffic to a database of known attacks. The warning can be delivered to the administrator as soon as an attack is detected or unusual activity is noticed. Installing an NIDS on the subnet where firewalls are to check for attempts to breach the firewall is an example of an NIDS in action.

- **Host Intrusion Detection Systems (HIDS):**

These programs run on separate hosts or other networked devices. Just the incoming and outgoing packets from the device are monitored by a HIDS, which notifies the administrator of any unusual or malicious behavior. It compares the current snapshot of the system files with the previous snapshot. An alert is given to the administrator to look into if the analytical system files were altered or deleted. Mission-critical equipment, which are not anticipated to modify their layout, are an example of HIDS utilization.

- **System for Protocol-based Intrusion Detection (PIDS):**

A system or agent that continuously remains at the front end of a server, regulating and interpreting the protocol between a user/device and the server, makes up a protocol-based intrusion detection system (PIDS). By continuously monitoring the HTTPS protocol stream and accepting the associated HTTP protocol, it tries to protect the web server. As HTTPS isn't secured and doesn't immediately enter the web presentation layer, the system would need to be located within this interface in order to use HTTPS.

- **Application Protocol-based Intrusion Detection System (APIDS):**

A system or agent that often lives within a server cluster is called an APIDS. By observing and analyzing communication on application-specific protocols, it detects intrusions. For instance, this would keep track of the SQL protocol that the middleware explicitly uses when communicating with the web server's database. To use HTTPS, m would need to be present in this interface.

- **Hybrid Intrusion Detection System:**

A hybrid intrusion detection system is created by fusing two or more intrusion detection system methodologies. Host agent or system data is merged with network data in the hybrid intrusion detection system to create a comprehensive picture of the network system. In compared to other intrusion detection systems, hybrid intrusion detection systems are more effective. Hybrid IDS is shown by Prelude.

1.5 IDS Detection Methods:

- **Technique based on signatures:**

An IDS that uses signatures to identify attacks looks for specified patterns in network traffic, such as the quantity of bytes or the proportion of 1s to 0s. Moreover, it identifies malware based on the well-known dangerous instruction sequence that it employs. The IDS's observed patterns are referred to as signatures. The assaults whose pattern (signature) is already present in the system may be quickly identified by signature-based IDS, but it might be challenging to identify newly discovered malware attacks since their pattern (signature) is unknown. [2]

- **Anomaly-based Method:**

As new malware is generated quickly, Anomaly-based IDS was invented to identify unknown malware threats. In anomaly-based IDS, machine learning is used to build a reliable activity model that is compared to anything arriving and is labeled suspicious if it is not found in the model. In comparison to signature-based IDS, machine learning-based methods have a machine learning-based methods have a superior generic characteristic since these models can be trained using different applications and hardware configurations. [2]

1.6 IDS evaluation measures

Measures that allow us to assess the overall effectiveness of detection systems :

- **Accuracy**: When an IDS system detects attacks without sounding false alarms, it is effective. When it labels a valid behavior in the environment as deviant or instructional, non-precision results[3]
- **Processing efficiency** is gauged by how quickly events are processed. Real-time detection will be feasible once the IDS system is more effective. [3]
- **Completeness:** An IDS's capacity to identify every assault. [3]
- **Fault tolerance:** The majority of intrusion detection systems use hardware or operating systems that are well-known to be weak points in an assault. Hence, an IDS should be able to withstand various assaults, notably denial of service assaults. [3]

For machine learning-based IDSs systems, a high detection rate is typically required to stop assaults before they result in any form of security breaches. The effectiveness of the Systems depends on having high detection accuracy and a low number of false alarms. While assessing the detection and categorization accuracy of assaults, the following factors should be taken into account are: [4]

- ✓ True Positive (TP): number of intrusions correctly detected
- ✓ True Negative (TN): number of correctly detected non-intrusions
- ✓ False Positive (FP): number of incorrectly detected non-intrusions
- ✓ False negative (FN): number of badly detected intrusions

A detector can make a variety of mistakes that in certain cases reduce or increase its power. True positives are situations where an alert is set off when a security policy is broken. The situations where no alarm goes off are the actual disadvantages. Triggers, but nothing unusual happens. False positives are situations where an alert sounds even if nothing out of the ordinary happens. False negatives are situations where an alarm does not sound even if something unusual is taking place. [5]

1.7 Conclusion

For information to be transmitted between multiple businesses in a secure and dependable manner, modern networked business settings demand a high level of security.

Once conventional technologies fail, an intrusion detection system serves as an adaptive backup technique for system security. Because cyber-attacks

will become more sophisticated, it is crucial that defense technology change to counter them. Here comes the importance of Intrusion Detection Systems.

Chapter 2:
Machine Learning for Cybersecurity

Chapter2: Machine Learning for Cyber-security

2.1 Introduction:

Artificial intelligence (AI)'s field of machine learning enables computers to learn without explicit programming. In order to offer robots the capacity to learn from data, analyze, and make well-informed decisions, researchers in this field seek to imitate human brain function as closely as possible, as well as the patterns of information processing and transmission seen in the organic nervous system. Several ICT products (such as image recognition, machine translation, medical diagnostics, etc.) as well as other diverse technical fields have effectively used machine learning approaches in recent years. (Autonomous car, intelligent robots, . . .etc).

But, the execution of the last mentioned is certainly based on the quality of the preparing information, they require a basic step called highlight building (Include Engineering), It is characterized as a strategy directed by domain specialists to choose vital highlights or properties information for each issue. For this, and with the accessibility of big-data, a unused process of machine learning called Profound learning (DL) was utilized to memorize the representation and verifiably abstracts the features. [6,7]

We began this chapter by discussing the value of deep learning in intrusion detection. Then, we provided a description and a categorization of different DL techniques. Additionally, we have described three DL strategies that will be the focus of our effort. We listed the various datasets used for the evaluation of IDSs systems based on machine learning at the conclusion of the chapter.

2.2 Machine Learning for Cyber-security

To deal with the challenges of cyber-security and the availability of large amounts of data related to cyber infrastructure, networks, operating systems, or information systems, methods and techniques like machine learning (ML), data mining, statistics, and other cross-disciplinary capabilities were used. The machine learning component deep learning may be employed. For IDSs that rely on anomaly detection or signatures. These approaches for categorizing and predicting cyber-attacks can be used to spot peculiar patterns and behaviors among different cyber-attacks, enabling a real-time cyber response. They are able to anticipate potential future assaults as well as recognize attacks when they have already occurred. Deep learning-based techniques can be used to overcome obstacles in the creation of a successful IDS. [8, 9]

Chapter2: Machine Learning for Cyber-security

2.3 Definition of deep learning:

Deep learning is a subset of machine learning which is simply a neural network with three or more layers.These neural networks make an effort to mimic how the human brain functions, however they fall far short of being able to match it, enabling it to "learn" from vast volumes of data. Additional hidden layers can assist to tune and improve for accuracy even if a neural network with only one layer can still produce approximation predictions.

Many artificial intelligence (AI) apps and services are powered by deep learning, which enhances automation by carrying out mental and physical activities without the need for human interaction. Deep learning is the technology that powers both established and upcoming technologies, like voice-activated TV remote controls, digital assistants, and credit card fraud. [10]

- ### 2.3.1 Functioning:

For any sort of these networks, the architecture of deep networks typically consists of an Input Layer, one or more Hidden Layers, and an Output Layer. Each pair of adjacent levels is linked. Weights are the relationships that exist between them. The "neurons" of the same layer, also referred to as "nodes," are unrelated. Figure 2.1 shows the typical design of a deep neural network model. [11]

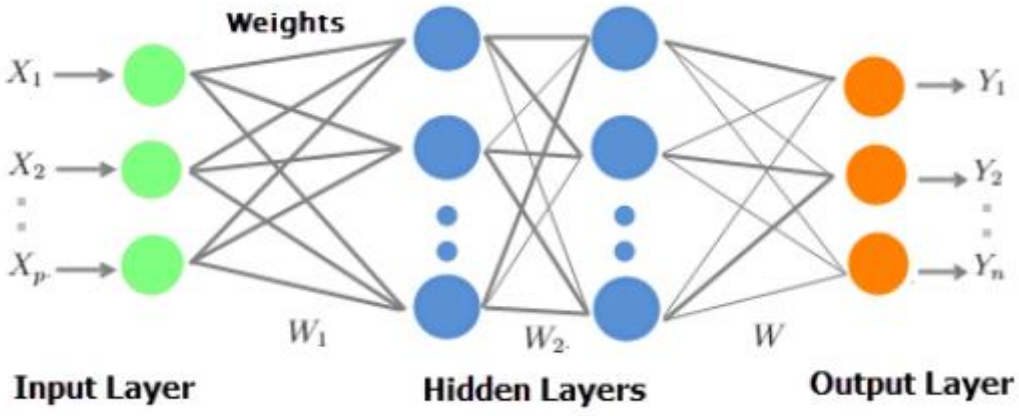


Figure 2.1: The architecture of a Deep Learning model. [11]

In order to create multiple levels of abstraction to represent data, deep learning uses a variety of machine learning techniques that use a flood of nonlinear neurons (nodes) arranged in multiple layers of processing. These techniques extract and convert entity variable values from the input vector.

- **Classification of DL methods:**

All deep learning techniques are actually neural networks, which have some basic characteristics in common. They are all layered structures made up of linked neurons. The network architecture, or how the neurons are arranged in the network, and occasionally how they are created, is what distinguishes them. Depending on how they are created and how they are meant to be used, these techniques may be divided into three models.

- When target label data is provided, deep discriminating models such as Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), and Convolution Neural Networks are employed for **supervised learning.** (CNNs).
- Deep learning is used for **unsupervised learning** when the input data is not labeled. Generative models, such as deep belief networks (DBN), deep auto encoders (DA), restricted Boltzmann machines (RBM), and deep Boltzmann machines, aim to group data according to certain similarity criteria for recognition or synthesis purposes. (DBM).
- **Deep learning hybrids:** a synthesis of the aforementioned models. Unsupervised deep networks may offer superior initialization as the foundation for testing discrimination (supervised learning).

2.4 Some deep learning methods:

Neurons are arranged in a series of linked layers to form deep neural networks. Their differences are based on the network's design, how the neurons are arranged within it, and how they act. Several deep learning model applications include:

2.4.1 Deep Neural Network (DNN):

A neural network with a specific degree of complexity is known as a Deep Neural Network (DNN), also known as Deep Nets. It may be thought of as stacked neural networks, or networks made up of numerous layers. Multilayer Perceptron's are a group of neurons arranged into a series of multiple layers. (MLP). By their depth and the quantity of layers and nodes (neurons) that make up the network, they differ from conventional neural networks (artificial neural networks). Deep neural networks are ANNs that include two or more hidden layers.

With combining several nonlinear transformations, they try to represent data with complicated topologies. Rosenblatt first presented the fundamental idea of perception in 1958. Their perception creates a linear combination based on its input weights (w), places the output via a nonlinear activation function, and then calculates a single output from numerous real-valued inputs (xi).

In mathematics, it may be expressed as When dealing with unlabeled and unstructured data, DDNs are frequently used. These advanced neural networks are now the go-to method for resolving a variety of computer vision challenges. [12]

$$y = \delta\left(\sum_{n=1}^{n} W_{ix_i} + b\right) = \delta(W^T X + b)$$

With:

- ✓ W: is the weight vector.
- ✓ X: is the input vector.
- ✓ b: designates the bias.
- ✓ δ: represents the activation function...

Typically, DNNs are employed to solve supervised learning issues. To create a model (learn it), all weights and biases must be adjusted to their ideal values.

2.4.2 Recurrent neural networks (RNNs)

RNNs were first created to help in sequence prediction; the Long Short-Term Memory (LSTM) algorithm, for instance, is renowned for its adaptability. inspired by how real neurons in the human brain work. These neurons are thought of as the center of contemplation, and occasionally they have to recall certain occurrences for use in the future before reaching a choice. Recurrent neural networks (RNNs) operate on the basis that humans reason using the knowledge that they have learned and that they have previously stored, which is a trait that traditional neural networks lack. These networks' fundamental design principle is the deployment of a recurrent calculation made possible by the architecture's loops. The network's output is a synthesis of its internal state (input memory) and the most recent input, while the internal state also adjusts to include the new input data. [13]

- ○ **Long Short Term Memory (LSTM):**

The RNN network has a large time step since it updates the weights while taking into consideration the previously saved state, the gradients became less and lower during training, and after a certain number of steps, the mistakes could not be transmitted farther along the network. There won't be a discernible difference in the outcome, thus the weights cannot be updated. Vanishing Gradients is the name of this RNN issue. German academics Sepp Hochreiter and Juergen Schmidhuber introduced a long-short-lived memory (LSTM) architecture for recurrent neural networks as well as extra stages known as Gated Recurrent Units (GRUs) in the mid-1990s to address this issue.

The performance and accuracy of RNNs have been enhanced using these procedures. The state of the cell is the central concept of the LSTM approach. It has the capacity to modify the cell state by removing or adding information. Structures known as gates control this method. The latter may take

the shape of a sigmoid function, where a value of 1 indicates that all information is delivered and a value of 0 indicates the reverse. quickly than LSTMs because it utilizes fewer training parameters and hence less memory. On datasets with longer sequences, LSTM is more accurate.

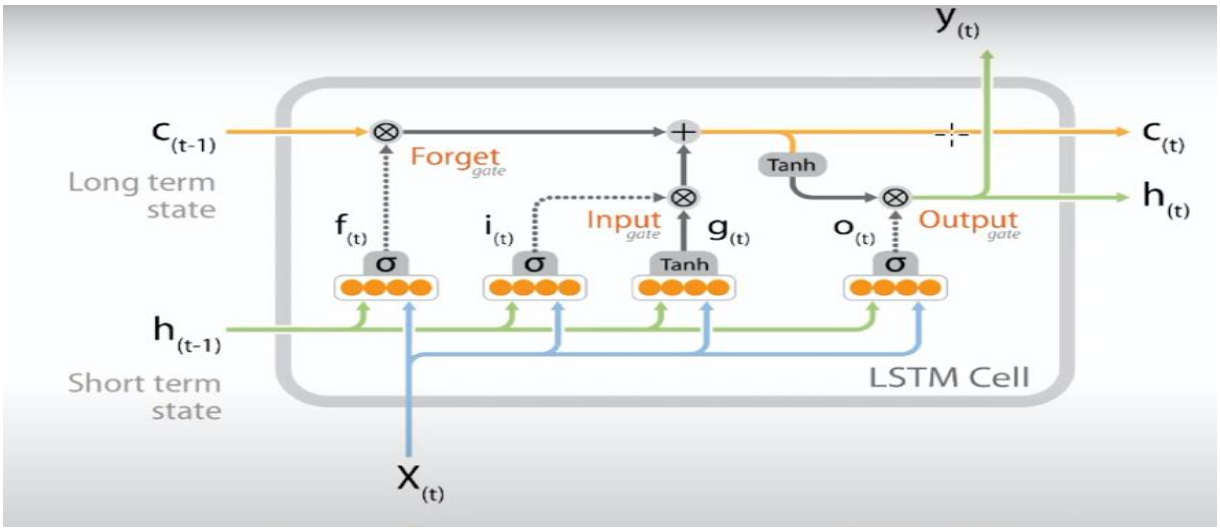


Figure 2.2: The architecture of an LSTM_GRU model

LSTM cells are the most effective in retaining useful information during gradient back propagation. This allowed them to correct the differences between the outgoing predictions and the reference categories by calculating the gradient of the error for each neuron, going from the last layer to the first. Figure 3.2 illustrates the four interactive layers (sigmoid and tanh), the three gates, and the Point wise operations that process vector x inside an LSTM cell at time t.

2.4.3 Convolutional neural networks (CNNs)

Convolution Neural Network, also known as CNN or ConvNet, is particularly adept at processing input with a grid-like architecture, like an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright it is and what shade it should be**.** [14]

Chapter2: Machine Learning for Cyber-security

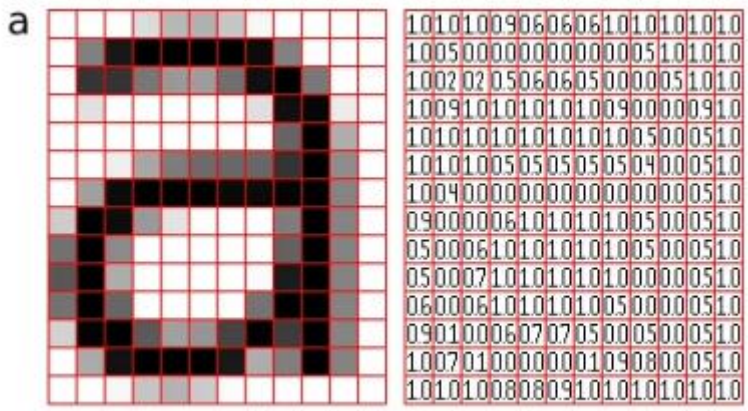


Figure 2.3: Representation of image as a grid of pixels. [15]

The traditional artificial neural network model, CNN, is a sophisticated variant with great potential. It is designed to handle increasing levels of complexity, preprocessing, and data compilation. It is based on the sequence in which neurons in the visual cortex of an animal's brain are arranged.

These were originally investigated for processing images in which repeating patterns can be found - for example, an image with repeating edges and other patterns. CNNs outperform all other classical ML algorithms and make great success in computer vision processing tasks (Computer Vision Tasks), they have wide applications in image and video processing, natural language processing (NLP), recommendation systems . . .etc.

Convolutional networks are particularly efficient thanks to several types of special layers: convolution layers , pooling layers and fully connected Figure 3.4 illustrates a model of a one-dimensional conventional network (1D CNN).

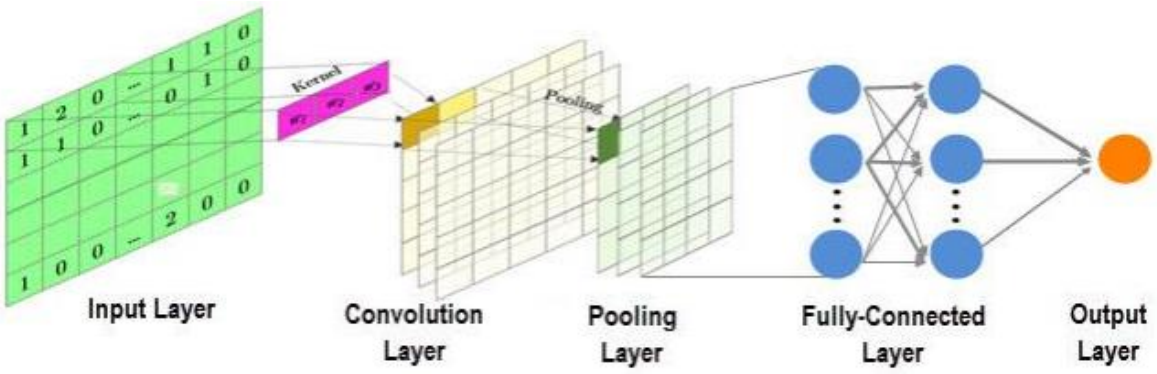


Figure 2.4: The architecture of a convolution neural network model

- **Convolution Neural Network Architecture:**

A CNN typically has three layers: a convolution layer, a pooling layer, and a fully connected layer.

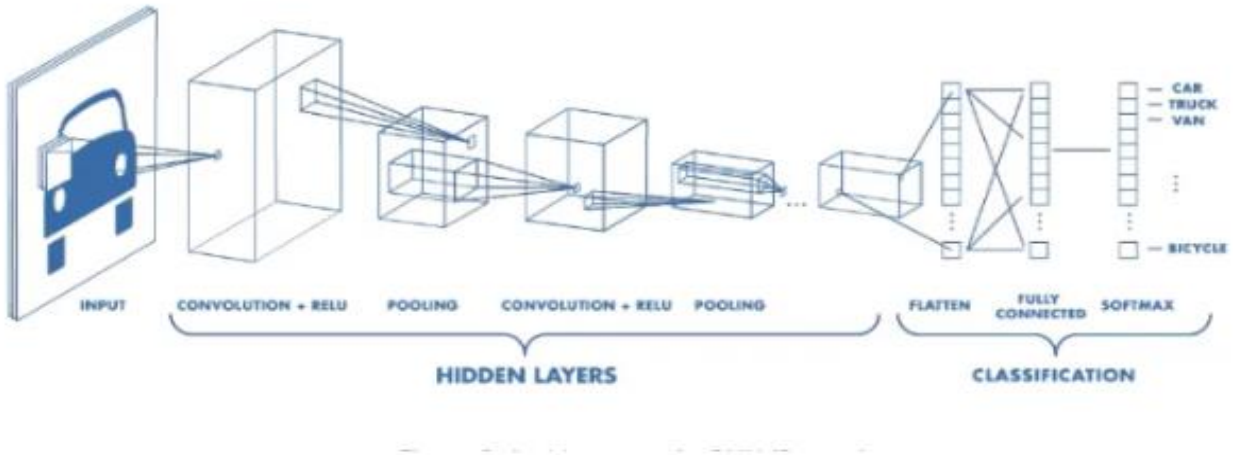


Figure 2.5: Architecture of a CNN

- **Convolution Layers:**

Extraction of high-level information is the aim of convolution. There are many learner filters (or cores) in it, and each one represents a different autonomous capability depending on the input volume. When passing forward (feed forward), each filter is convolved across the width and height of the input volume, calculating the product of the points between the inputs and the filter values to produce a new feature map that more accurately depicts the data. These filters are made up of a layer of connection weights, have a small receiving field (the size of the kernel), and a small transmitting field (the size of the input volume). As a result, the network gains knowledge about the filters that turn on when detects a type of feature that is important and specific to a certain spatial position in the entrance.

- **Pooling Layer**

By calculating an aggregate statistic from the surrounding outputs, the pooling layer substitutes for the network's output at certain points. This aids in shrinking the representation's spatial size, which lowers the using this network paradigm. The group of neurons from the preceding layer are controlled by the neurons of the convolution layers.

A weighted average based on the distance from the center pixel is one of the pooling functions, along with the average of the rectangular neighborhood and the L2 norm of the rectangle neighborhood. However, max pooling, which returns the highest output from the neighborhood, is the most widely used technique.

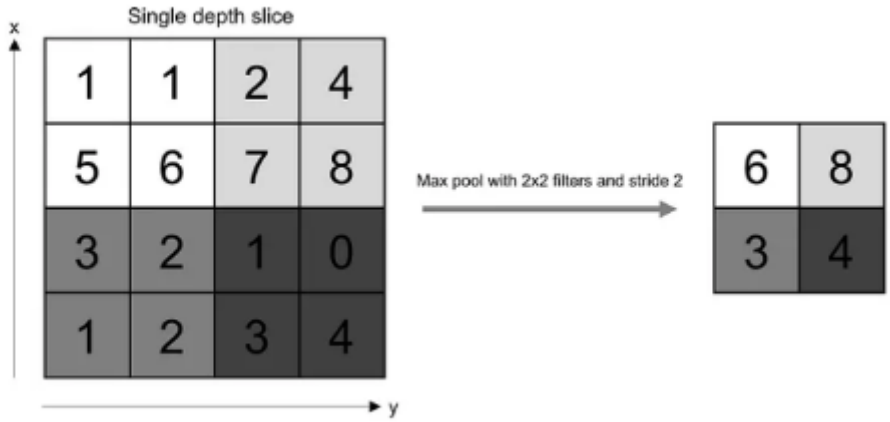


Figure 2.6: Pooling Operation.

If we have an activation map of size *W* x *W* x *D*, a pooling kernel of spatial size *F*, and stride *S*, then the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F}{S} + 1$$

This will yield an output volume of size *Wout* x *Wout* x *D*.

In all cases, pooling provides some translation invariance which means that an object would be recognizable regardless of where it appears on the frame.

- **Fully Connected Layer**

Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect.

The FC layer helps to map the representation between the input and the output.

- **Non-Linearity Layers**

Since convolution is a linear operation and images are far from linear, non-linearity layers are often placed directly after the convolution layer to introduce non-linearity to the activation map.

There are several types of non-linear operations, the popular ones being:

- Sigmoid

The sigmoid non-linearity has the mathematical form $\sigma(\kappa) = 1/(1+e^{-}\kappa)$. It takes a real-valued number and "squashes" it into a range between 0 and 1.

However, a very undesirable property of sigmoid is that when the activation is at either tail, the gradient becomes almost zero. If the local gradient becomes very small, then in back propagation it will effectively "kill" the gradient. Also, if the data coming into the neuron is always positive, then the output of sigmoid will be either all positives or all negatives, resulting in a zig-zag dynamic of gradient updates for weight.

- Tanh

Tanh squashes a real-valued number to the range [-1, 1]. Like sigmoid, the activation saturates, but — unlike the sigmoid neurons — its output is zero centered.

- ReLU

The Rectified Linear Unit (ReLU) has become very popular in the last few years. It computes the function $f(\kappa)=\max(0,\kappa)$. In other words, the activation is simply threshold at zero.

In comparison to sigmoid and tanh, ReLU is more reliable and accelerates the convergence by six times.

Unfortunately, a con is that ReLU can be fragile during training. A large gradient flowing through it can update it in such a way that the neuron will never get further updated. However, we can work with this by setting a proper learning rate.

2.5 Conclusion

Deep learning proves to be valuable. All that's required are plenty of computing power and training datasets. It is expanding quickly; new designs, methods, or versions develop every week. An important research path and emphasis for security researchers continues to be the use of novel DL techniques and assessing the performance of several current DL structures.

Chapter 3:
CNN based Intrusion Detection System

Chapter3: CNN Based Intrusion Detection System

3.1 Introduction

Nowadays everything is connected directly to the Internet. So whenever you are online there is always a possibility of an attack which poses a security risk.

Security professionals and engineers are paying more and more attention to identify network attacks, trying to find solutions for governments and private organizations to protect their information assets and prevent detection of illegal or unwanted intruders. Classification-based IDS is designed to classify network traffic into two classes, "General" and "Intrusion".30 years ago, various researches using machine learning methods have been studied, nonetheless the false alarm rate is high and the detection is low, which affects the IDS solution. Deep learning (a branch of AI) has been widely used in fields of classification and pattern recognition, it applies various levels of information processing layers within a hierarchical design .Among the various deep learning algorithms, CNN is efficient in complex tasks like identifying faces and objects . So, many researchers attempted to use CNN to solve the IDS mystery, or use CNN to solve the IDS puzzle. In this chapter we present a description of the chosen datasets and CNN –Based IDS in detail.

3.2 Taxonomy of CNN based IDS

Figure 3.1 below shows the taxonomy of CNN-based IDSs, based on the D environment and the methods that had been divided into

First level Applications include Internet of Things (IoT), NIDS, Software Defined Networks (SDN), In-Vehicle Networks , Smart Homes, Cloud Security, Advanced Metering Infrastructure (AMI), Spark Platform, Wireless Networks, Industrial Control System "ICS".

Second level, single and hybrid methods, each one has its work selected, about hybrid methods: one or more machine learning, deep learning or soft learning CNN calculation methods can be combined , a single or hybrid CNN can be applied to any environmental application. Unlike other methods like resizing and dial functions, classified as a single CNN. For example, hybrid LSTM means the authors used hybrid CNN-LSTM, but single PCA means the authors use a CNN and a PCA for dimension reduction Each method of each category has its own work selected as seen in the figure.[16]

Chapter3: CNN Based Intrusion Detection System

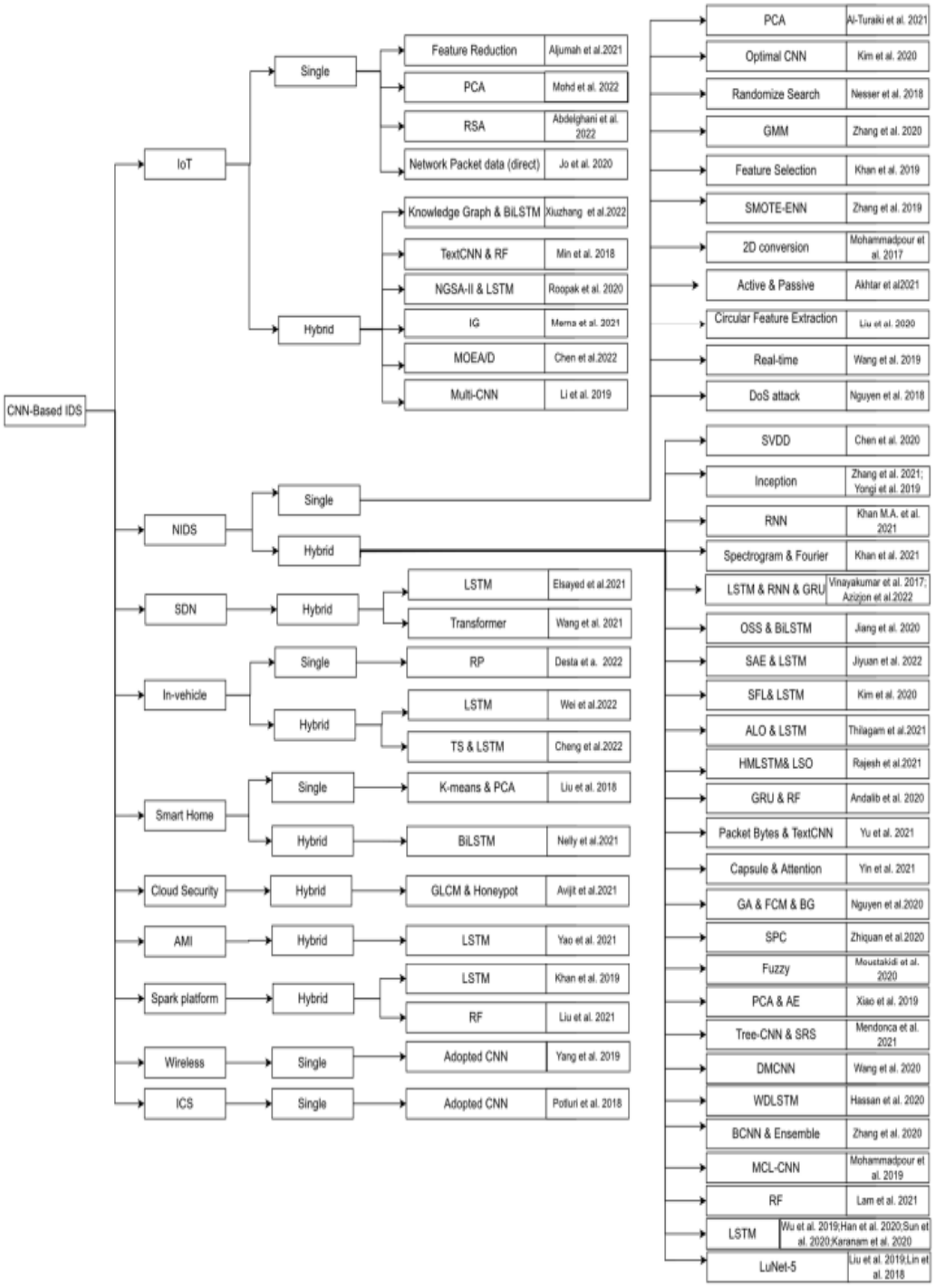


Figure 3.1: Taxonomy of CNN-based IDSs [16]

Chapter3: CNN Based Intrusion Detection System

3.3 Performance Metrics

A visual interpretation of how machine learning algorithms work is not possible, which means that quantitative metrics (e.g. PPV, TPR, F1 and confusion matrix) must be used to run evaluations. As with other classification problems, IDS task confusion Matrix contains terminologies such as:

- **True-Positive (TP)** refers to an attack data classified correctly as an attack.
- **False-Positive (FP)** refers to misclassified normal data as an attack.
- **True-Negative (TN)** refers to normal data classified as normal and **false negative** (FN) refers to attack data incorrectly classified as normal

Table 1 shows the terminology and performance assessment measures used in the IDS studies. For more details, the table lists all terminologies used for the same measure.

Terminology	Explanation	Equation
ACC	Accuracy	$\frac{TP + TN}{TP+TN+FP+FN}$
PVV	Precision and positive predictive value	$\frac{TP}{TP+FP}$
TPR	Recall, sensitivity, true positive rate and detection rate	$\frac{TP}{TP+FN}$
FNR	Miss rate and miss false negative rate	$\frac{FN}{TP+FN}$
TNR	Specificity, selectivity and true negative rate	$\frac{TN}{FN+TP}$
FPR	False alarm and false positive rate	$\frac{FP}{TN+FP}$
F1Score	F1Score and F measure	$\frac{2*PPV*TPR}{PPV+TPR}$
ROC	ROC curve: a graph with the x-axis shows FPR and y-axis shows TPR	
AUC	The area under the curve or area under the Roc curve	

Table 3.1: The equations and explanations of each measure.

3.3.1 Accuracy:

Accuracy is the most important and the simplest measure as well . The accuracy is a percentage determined by the number of correct predictions divided by the number of all predictions (multiplied by 100)[17]. A significantly high percentage means that the model predicts well, a low

value means the model is underperforming. A percentage close to 50% on a binary classification means that the model in results close to a random assignment of classes, so it is useless, if we have an imbalance, accuracy is not a suitable measurement technique so it can't provide valid information so using other relevant assessment indicators such us precision recall f1score can be helpful.

3.3.2 Precision, Recall and F1Score:

Precision measures the number of correct classifications penalized by the number of incorrect classifications[18]. Meanwhile **Recall** or sensitivity measures the number of successful classifications penalized by the number of missed entries and the false positive rate measures the percentage of events incorrectly classified as harmful .

The **F1Score** result calculates the harmonic mean of precision and recall, which serves as the derived measure of performance.

3.4 CNN Based IDS approaches:

• CNN-based IDS schemes can be categorized into four groups:

1. CNN only
2. CNN and RNN
3. CNN and other deep learning methods
4. CNN and machine learning , fuzzy, Fourier transformation, clustering, or evolutionary algorithms

• Tables 4.2–4.3 provide essential information to compare CNN-based IDS schemes, such as "year of publication", "architecture", "simulators/environments", "input layer shape", "datasets used for training and evaluation", "performance", "feature extraction method", "classification method", "performance metrics", and "other methods used alongside the CNN"

• Notations for the "simulators/environments" column in the Tables include: KT – Keras using the TensorFlow backend; KH – Keras using the Theano backend; T – TensorFlow; M – MATLAB; S – Scikit-learn; J – Java with Deeplearning4j; W – WEKA; and P – PyTorch [16]

3.4.1 Single CNN-Based Schemes:

- Research that focuses on original and improved versions of CNNs without incorporating any other machine learning or deep learning algorithms

• Al-Turaiki et al. proposed a two-step preprocessing strategy combining dimensionality reduction and feature engineering

• Lam et al. employed statistical behaviors rather than typical anomalous attack behaviors

• Jo et al suggested three preprocessing methods including “direct”, “weighted”, and “compressed”

• Kim et al. examined how different attacks within the same category can be detected[16,19]

• Table 4.2 compares the experimental information, including the authors, the year of publication, ENV, datasets, and evaluation performance

• Table 4.3 compares architectural properties such as architecture, input layer shape, feature extraction method, and classification method

Authors	Year	ENV	Datasets	Binary ACC(%)	Multiclass ACC(%)	Evaluation Metrics
Mohammadpour et al	2019	T	CICIDS2017	99.87	___	ACC,PPV,TPR,F1Score,FPR
AL-Turaiki et al	2021	KT	NSL-KDD UNSW-NB15	90.14 89.26	81.44 68.25	ACC,PPV,TPR,F1Score
Lam et al	2021	___	CSE-CIC-IDS2018	99.99	99.98	ACC,PPV,TPR,F1Score
Jo et al	2020	T	NSL-KDD	88.82	___	ACC,PPV,TPR,F1Score
Kim et al	2020	T	KDD Cup99 CSE-CIC-IDS2018	99.00	91.50	ACC,PPV,TPR

Table 3.2 Properties of group 1 (Single CNN-Based) IDS schemes.[16]

Authors	Architecture	Input Shape	Feature Extraction	Classifier
Al-Turaiki et al	5Conv,2MaxPool,4FC	2D(11×11) 2D(14×14)	CNN	SoftMax
Lam et al	3Conv,2MaxPool,2FC	1D	CNN	SoftMax
Jo et al	1Conv,1MaxPool,1FC	2D(28×28)	CNN, Weighted & compressed	SoftMax
Kim et al	3Conv,2MaxPool	2D(13×9) 2D(13×9×3)	CNN	SoftMax
Aljumah et al	2Conv,1MaxPool,1BatchNo,2FC,1Dropout	1D	CNN, attribute transformation and reduction	Softmax

Table 3.3: Architectural properties of group 1 (Single CNN-Based) IDS schemes.[16]

3.4.2 Hybrid CNN and Deep Learning Schemes

• Multiple models inspiring by well-known deep learning architectures (i.e., Inception and TextCNN)

• Models combine deep learning algorithms, such as autoencoder (AE) and denoising autoencoder (DAE), to denoise data, clean data, and extract features

• Two-step network intrusion detection system must developed by based on likeGoogLeNet Inception and CNN models

• CNN algorithm in incorporates a batch normalization algorithm and inception model to train the system and enhance the model's convergence speed

• Deep capsule network-based IDS on an attention mechanism proposed by Yin et al. to make the model focus on features with high impacts[16]

3.4.3 Hybrid CNN and Other Machine Learning Method Schemes

• a system has developed which combines IDS data into a single actionable risk indicator

• The system consists of a fuzzy allocation scheme, a Vec2im transformation mechanism, and a dimensionality reduction module

• and proposed genetic algorithm-based exhaustive search and fuzzy C-means clustering to identify bagging classifiers

• also proposed a Tree-CNN hierarchical algorithm with a soft-root-sign activation function to reduce the time needed for a generated model to be trained and detect DDoS, infiltration, brute force and web attacks

• The model is simple and requires less process time and fewer calculation tools, making it more efficient than other existing IDSs that use machine learning algorithms[16]

Chapter3: CNN Based Intrusion Detection System

3.5Analytical Investigation

This section lists issues with CNN-based IDS systems. Therefore, it is important to present some statistics for illustration questions like:

1. Annual Percentage of CNN-IDS Articles Published.
2. Percentage of IDS approaches with 1D and 2D inputs as input Dimensions CNN.
3. Percentage of IDS solutions used in different survey datasets rate performance.
4. Percentage of use of proven implementation frameworks IDS schemes.
5. The number of convolution layers added in CNN-based models showing depth pattern.
6. Percentage of each evaluation metric used to test IDS schemes.
7. Accuracy of IDS approaches tested on specific data sets

These statistics are available for 66 CNN-based IDS approaches **Figure 3.2** shows that researchers have developed and tested many programs based on CNN Approaches to Solving the IDS Problem. From all studies reviewed of these works, 3% were published in 2017 and 28.8% in 2021. It is also, note that all articles selected for 2022 were published before July 2022.This clearly shows that research interest in this area is growing rapidly

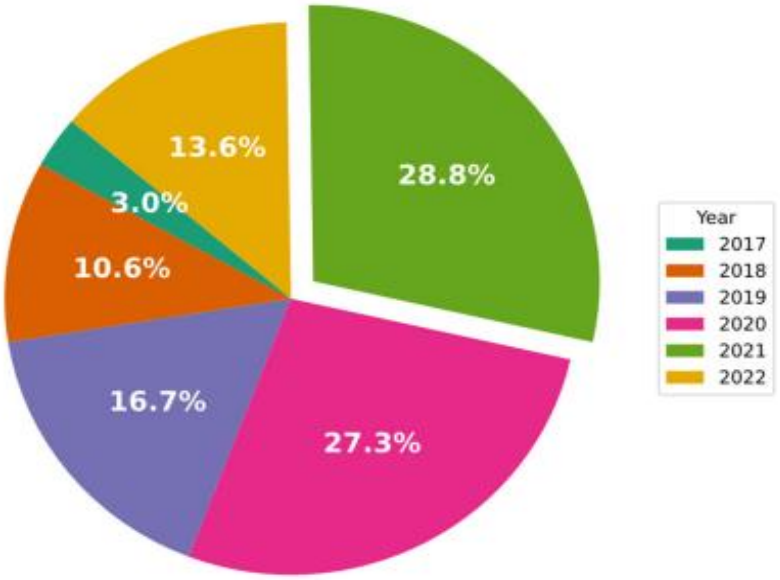


Figure 3.2: Percentages of CNN-IDS papers published each year.[16]

Chapter3: CNN Based Intrusion Detection System

CNN specializes in processing data with a grid-like architecture (e.g. images)[18]. Therefore, many CNN-IDS studies have converted the input features to a matrix shape (2D) form. In addition, several studies have used 1D data properties as input to CNN models. Using an image format has been found to reduce the number calculated parameters needed in the CNN model. A one-dimensional array can be turned into a two-dimensional (2D) or three-dimensional (3D) image. In several studies, a 49-feature vector was transformed into a 7 × 7 matrix .If the number of features is not squared, you can add zeros to the feature vector. On the other hand, researchers tried to fill eight gaps in the matrix with zero values random cells (this was possible because the initial set of features had a dimensionality of 41, with a total of 49 fields to fill in the matrix).

→ **Figure 3.3** shows the percentage of IDS methods using 1D and 2D input forms to achieve this. It also displays the fact that 63.8% of the methods use 2D inputs while only 36.2% use 1D inputs.

→ In **Figure 3.4**, the datasets used in the examined CNN-based IDS approaches studies were presented. The figure shows that 33% of IDS approaches still use NSL-KDD,, while 14.9% use the older KDD Cup99 dataset. In addition, in some cases, multiple datasets were used to conduct a more comprehensive assessment of the proposed approaches. Total 14.9% of the studies included in this review analyzed the CIC-IDS2017 dataset.

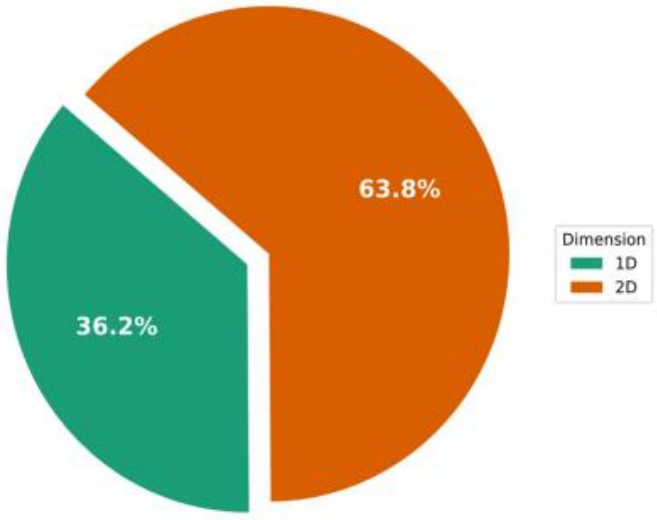


Figure 3.3: Percentages of IDS schemes that have used 1D or 2D (image) input shape [16]

Chapter3: CNN Based Intrusion Detection System

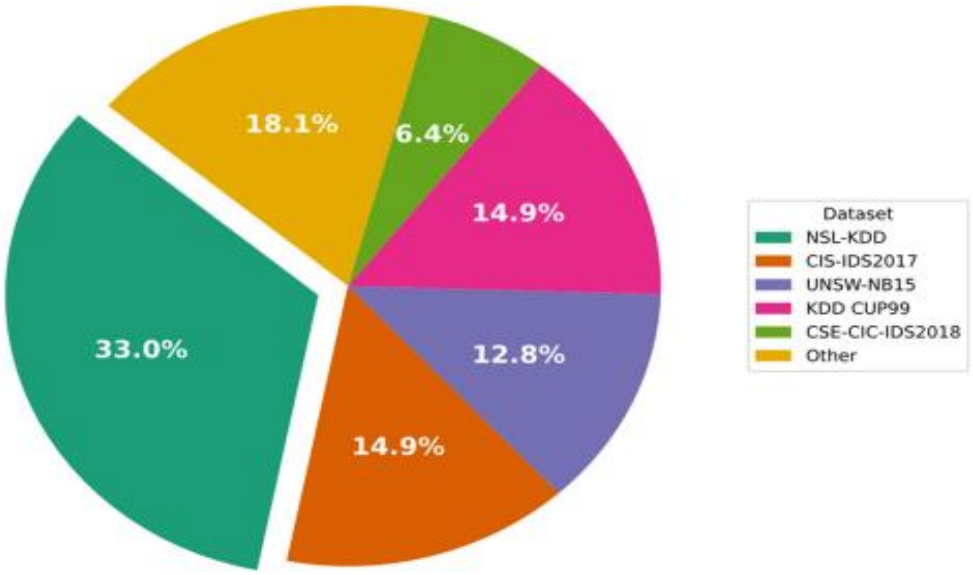


Figure 3.4: Percentages of datasets applied in CNN-IDS solutions [16].

the number of convolutional layers used by IDS schemes is displayed in **Figure 4.5** . It shows two convolution layers were used in 32 approaches. In addition, ten IDS approaches appear to use more than three layers of convolution.

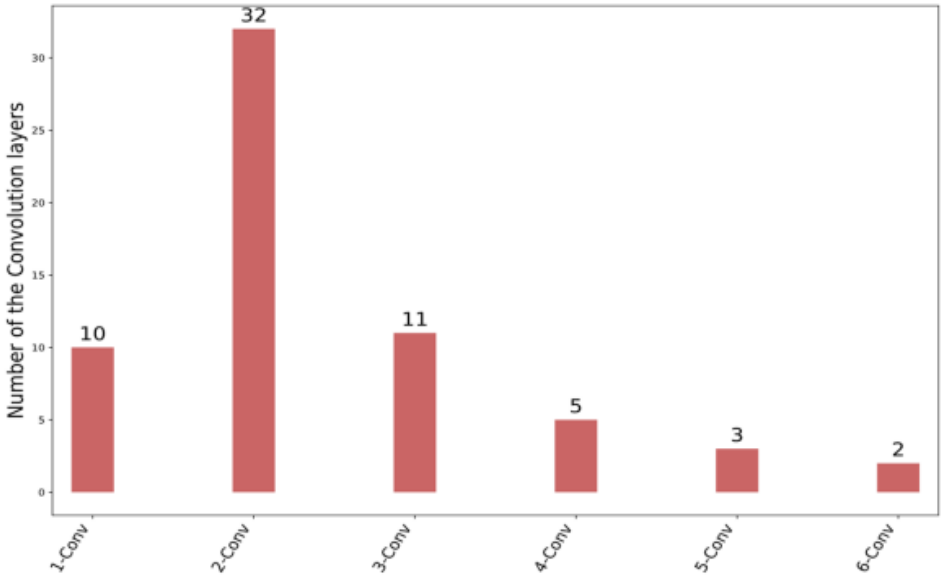


Figure 3.5: Number of convolutional layers that have been used for IDS schemes [16]

Figures 3.5 to 3.8 are to describe the accuracy of the CNN-based IDS schemes examined here using these datasets: NSL-KDD, CIC-IDS2017 .

In these figures, the colors green and red show the precision above or below 95%. To allow comparison of the results, the results for each record are presented individually. Researchers can compare these results with their own results. Has the maximum value between binary and multiclass classifications been chosen for the storyline in all characters?

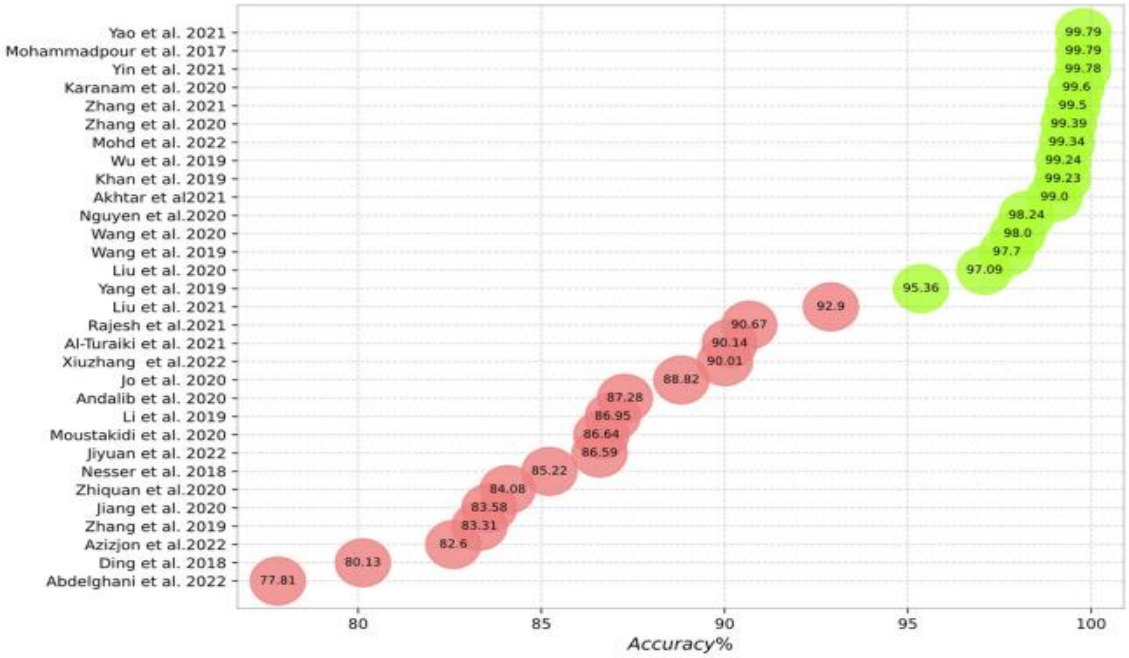


Figure 3.6: Accuracy of the studied schemes on the NSL-KDD dataset.[16]

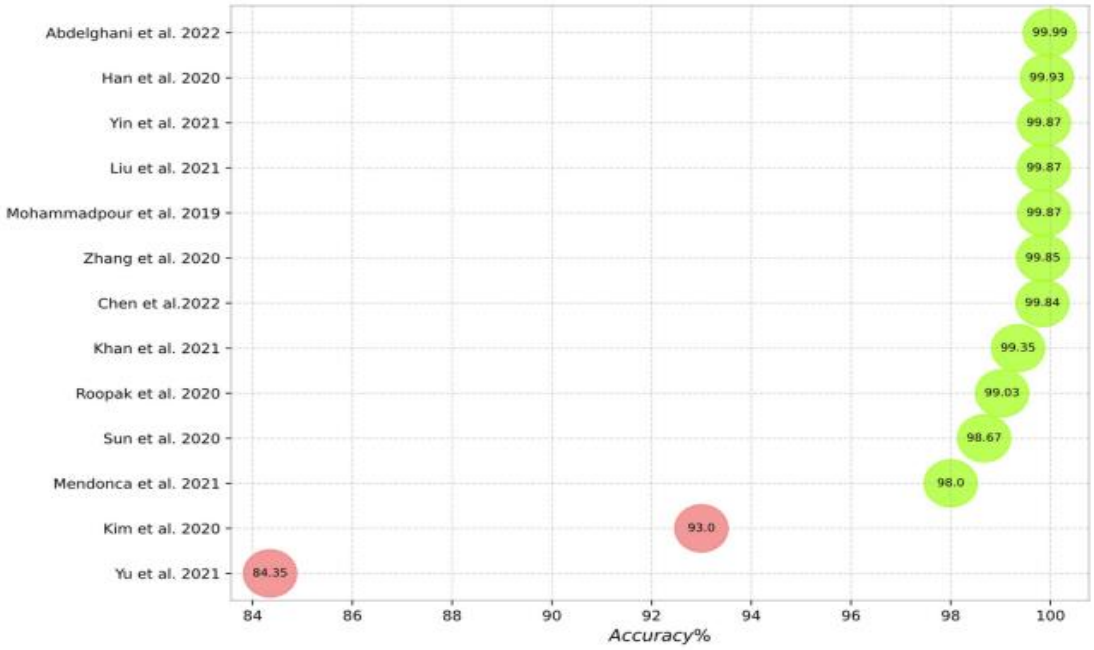


Figure 3.7: Accuracy of the studied schemes on the CIC-IDS 2017 dataset [16]

Chapter3: CNN Based Intrusion Detection System

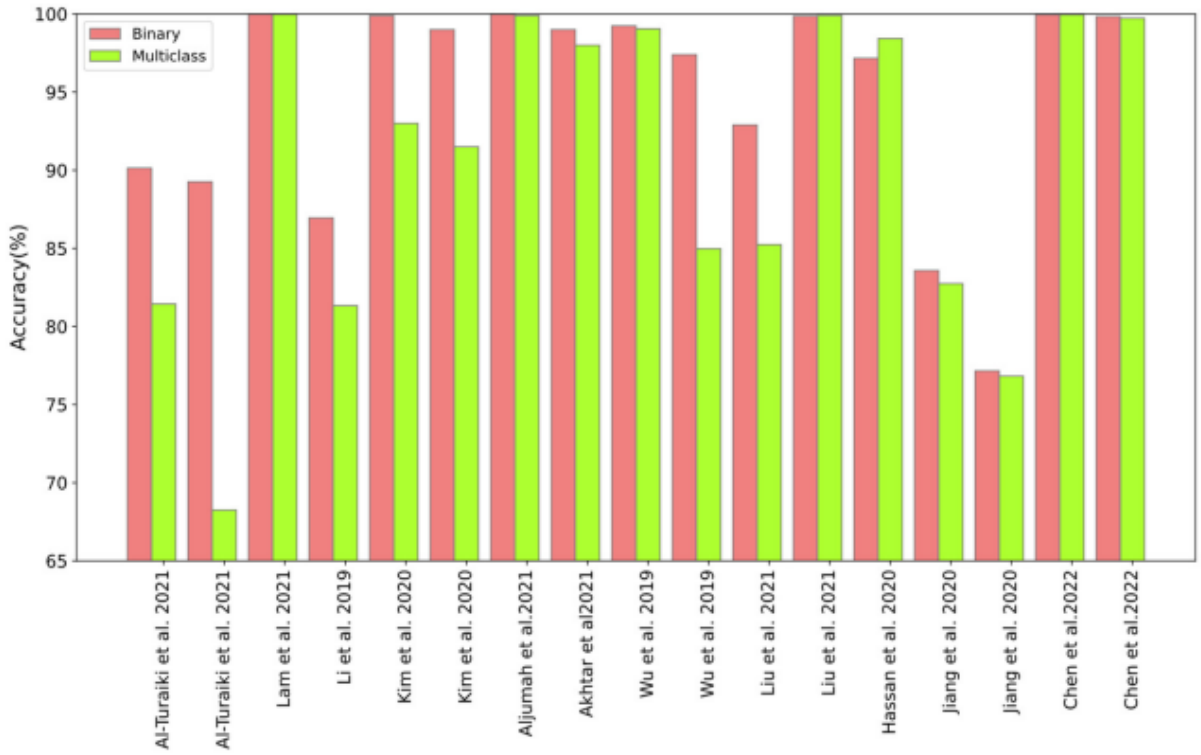


Figure 3.8: Comparison between binary and multiclass classification of the CNN-based IDS [16]

3.5 Conclusion

IDSs leverage a variety of AI approaches, such as machine learning, to improve performance against emerging cyber-attack challenges. They play a significant role in protecting networks and computer systems throughout the world. Because it can independently find relevant features in large amounts of data, deep learning has a significant advantage over other traditional machine learning techniques. Researchers frequently employ the CNN deep learning algorithm to enhance IDS solutions with relation to privacy concerns and security dangers. As a result, this chapter offers a thorough analysis of CNN-based IDS methods. Additionally, prevalent primary datasets used by IDS schemes are demonstrated in this paper. Four different categories are used to organize the various approaches. We describe the general structures of the learning techniques, datasets, data input forms, performance measures, feature extraction techniques and classifiers.

Chapter 4:
Implementation, Results and Discussion

Chapter 4: Implementation, Result and Discussion

4.1 Introduction

Previously in Chapter 4 CNN-based IDSs was described in theory, but in this chapter the idea will be presented from the implementation side, listing the tools, functions, models, datasets used and test protocols till goal the of our study is affirmed and valid in results as well , a set of supervised measures such as recall, precision, accuracy, f1score and loss function been used . To refine the results, a whole scenario of a comparative study with different datasets, models, and test sizes is employed to improve model performance.

4.2 Choice of programming language

Python is a general purpose interpreted programming language. It is easy to learn and use primarily because the language focuses on readability[20]. Its built-in high-level data structures combined with dynamic typing and dynamic linking make it very attractive for rapid application development as well as for use as a scripting or linking language to link existing components. Python's simple and easy-to-learn syntax emphasizes readability and reduces program maintenance costs[21]. Python supports modules and packages that promote program modularity and code reuse. The Python interpreter and extensive standard library are freely available in source or binary form for all major platforms and are free to redistribute.

4.3 Implementation Tools

An open source deep learning framework was applied at simplifying the implementation of complex and large models and that to implement and optimize the model we offer.

- **TensorFlow :**

  TensorFlow is a comprehensive open source platform for building machine learning applications, it is a symbolic math library that uses data flow and differential programming to perform a variety of tasks focused on training and deep neural networks inference. Allows novices and experts alike to build machine learning models for desktop, mobile, web, or cloud using a variety of tools, libraries and community resources . and have multiple wrappers in multiple languages such as Python, Java , C++. The most popular uses of TensorFlow(its applications.):video detection ,image and voice recognition and text apps .

- **Keras**

Keras is an API designed for humans, not machines. Keras adheres to best practices to reduce cognitive load: it offers consistent and simple APIs, minimizes the number of user actions required in common use cases, and provides clear and helpful feedback when user errors occur[22]. It also includes extensive documentation and development guides. Keras solve many different problems like image classification using different architectures such as VGG16,Xception,ResNet ,Inception…[23]

- **SkLearn**

Sklearn (scikit-learn) is a Python library that provides a variety of supervised and unsupervised machine learning algorithms. It's based on some technologies like NumPy, Pandas and Matplotlib. The main use cases of this library can be divided into the following 6 categories:

The main use cases of this library can be divided into the following 6 categories:

- Preprocessing (min-max normalization)
- Regression (logistic regression)
- Classification (CNN)
- Clustering (K-means)
- Model Selection
- Dimensionality Reduction

- **Google Colaboratory**

Google Colab or Colaboratory is a cloud service provided by Google (free).It is based on Jupyter Notebook and 0 intended for learning and research automatically. This platform allows you to train machine learning models directly in the cloud, with: No configuration required, free access to GPU and Easy Sharing

4.4 CNN model architecture

In a traditional neural network, data is first passed to the input layer. Then it moves to the hidden level and exits from there Layer. The layers are fully connected and there is no connection between nodes of the same layer. A conventional neural network therefore poses many unsolvable problems. Its architecture is an improvement of Standard Neural Network Architecture. As a result, CNN has achieved remarkable results in areas such as image classification and language analysis.

Chapter 4: Implementation, Result and Discussion

On CNN are:

(1) One or more convolutional layers
(2) Pooling layers at the top
(3) Fully connected layers
(4) Dropout layers serving as regularization layers

With this structure, CNN can take advantage of the two-dimensional structure of the input data. Thus, the network can use an image as input. Thereby we avoid the complicated feature extraction and unnecessary data reconstruction of traditional recognition algorithms. The modeling power can be increased and so that the level of difficulty of increased manual processing of data can be reduced through bundling, shared weights, and infrequent connectivity. CNN can learn from a vast amount of unlabeled data at varying levels of functionality. Therefore, the possible uses of CNN in areas such as network intrusion detection are versatile. The architecture of a method similar to the one proposed is shown in **Figure 4.1**

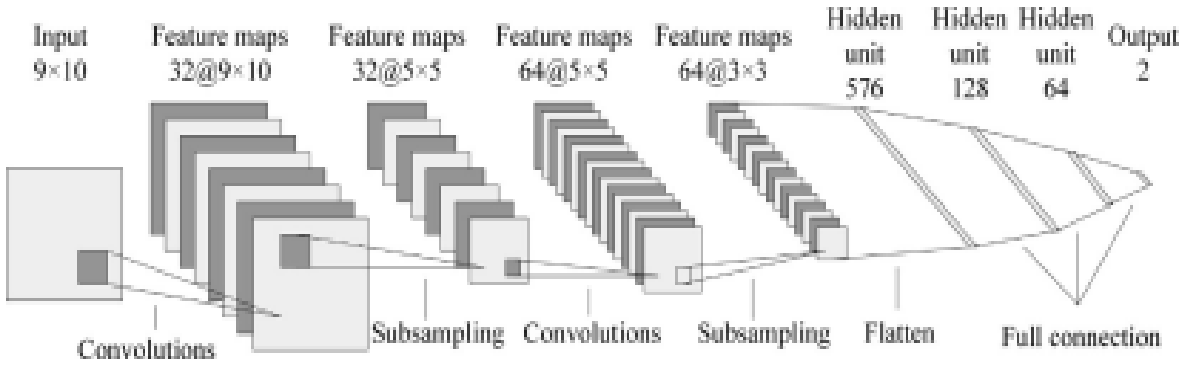


Figure 4.1: An example of the architecture of a single CNN model

In Python programming, the sequential type is the most commonly used pattern type. This is the easiest way to create a CNN model in Keras. This allows us to build up the model layer by layer. Layers are added to the model using the add () function. For this model, there are three Convolution and three Pooling followed by a Flatten layer which is usually used as a connection between Convolution and the Dense layers. The Dense layers are often used for the output layers. The activation employed is 'Softmax' which gives the probability for each class and gives a total of 1. The model makes a prediction based on the class with the highest probability.The summary of the model is displayed in the table 4.1 below

Chapter 4: Implementation, Result and Discussion

Layer(type)	Output Shape	Param #
input_1 (InputLayer)	(None, 9, 13, 1)	0
conv2d (Conv2D)	(None, 9, 13, 100)	500
max_pooling2d (MaxPooling2D)	(None, 4, 6, 100)	0
conv2d_1 (Conv2D)	(None, 4, 6, 50)	20050
max_pooling2d_1 (MaxPooling2D)	(None, 2, 3, 50)	0
flatten (Flatten)	(None, 300)	0
dense (Dense)	(None, 10)	3010
dense_1 (Dense)	(None, 15)	165

Table 4.1: the summary of the CNN

4.5 Datasets Used

- **NSL-KDD**

In order to train and test an intrusion detection system, A large dataset with a large amount of high-quality data that mimics real time was needed, the NSL-KDD dataset which is an advanced version of the previous KDD99 dataset. There are four files contained in this dataset, two of which are for training ("KDDTrain+" and "KDDTrain_20%") and two of which are for testing ("KDDTest+" and "KDDTest-21") . The NSL-KSS training and testing datasets are shown in Table 4.2.[16]

File	Description	Num of Samples	Num of Normal
Train+	Full training set	125,973	67,343
Train20Percent	20%of the training set	251,92	13,499
Test+	Full testing set	22,544	9711
Test-21	A subset of training set	11,850	2152

Table 4.2: NSL-KDD training and testing files

In this project, the NSL-KDD dataset is operated to study the efficiency of the classification algorithm in detecting anomalies. It has 42 attributes, which means there are no redundant instances as KDD99 and better classification results. The attribute number42 in the dataset is the 'class' attribute which specifies whether the given instance is a normal connection or an attack by giving the name of attack.

Chapter 4: Implementation, Result and Discussion

The following table 4.3 describes the attributes of the dataset with an explanation of each one

N	Features Name	Features Description
1	duration	Duration of connection in sec
2	protocol_type	Connection protocol(tcp, udp, icmp)
3	service	Service type (e.g. http, ftp …)
4	flag	Normal or error status flag of connection
5	src_bytes	The number of bits from src to dst
6	dst_bytes	The number of bits from dst to src
7	land	1 if con is from/to the same host/post;else0
8	wrong_fragment	Number of wrong fragment (values 0,1,2)
9	urgent	Number of urgent packets
10	hot	Number of hot indicators
11	num_failed_logins	Number of failed login attempts
12	logged_in	1 if successfully logged in; else 0
13	num_compromised	Number of compromised conditions
14	root_shell	1 if root shell is obtained; else 0
15	su_attempted	1 if ‘su root’command attempted; else 0
16	num_root	Number of ‘root’ accesses
17	num_file_creations	Number of file creation operations
18	num_shells	Number of shell prompts
19	num_access_files	Number of operations on access control files
20	num_outbound_cmds	Number of outbound cmds in an ftp session
21	is_host_login	1 if ‘login’ belongs to ‘hot’list ; else 0
22	is_guest_login	1 if ‘login’ is a ‘guest’ login; else 0
23	count	Number of connecting same hosts in past 2s
24	srv_count	Number of connecting same services in past 2s
25	serror_rate	% of connections that have ‘SYN’ errors
26	srv_serror_rate	% of connections that have ‘SYN’ errors
27	rerror_rate	% of connections that have ‘REJ’ errors
28	srv_rerror_rate	% of connections that have ‘REJ’ errors
29	same_srv_rate	% of connections to the same services
30	diff_srv_rate	% of connections to the different services
31	srv_diff_host_rate	% of connections to the different hosts
32	dst_host_count	Number of connecting same host
33	dst_host_srv_count	Number of same host and same service
34	dst_host_same_srv_rate	Rate of same host and same service
35	dst_host_diff_srv_rate	Rate of different service in different host
36	dst_host_same_src_port_rate	Rate of connecting host in same src port
37	dst_host_srv_diff_host_rate	Rate of connecting host in different src port
38	dst_host_serror_rate	% of connections to current host that have SO error
39	dst_host_srv_serror_rate	% of connections to current host and specified service that have an SO error
40	dst_host_rerror_rate	% of connections to current host that have RST error
41	dst_host_srv_rerror_rate	% of connections to current host and specified service that have an RST error
42	label	Classified to normal and abnormal (38 different label in our dataset

Table 4.3: NSL KDD dataset attributes

Chapter 4: Implementation, Result and Discussion

Attacks are presented in this dataset by 4 classes: DoS, Probe, R2L, U2R, the following table4.4 presents the type of each attack class and sample relevant feature, with an example for each, while table 4.5 shows the number for each one.

Attack class	Attack Type	Sample Relevant Feature	Example
DoD	Apache2, Back, Pod, Process table, Worm, Neptune, Smurf, Land, Udpstorm, Teardrop	Percentage of packets with errors – source byte	Syn flooding
Probe	Satan, Ipsweep, Nmap, Port sweep, Mscan, Saint	Source bytes – duration of connection	Port scanning
R2L	Httptunnel, Snmpgetattack, Guess-Password, Imap, Warezmaster,Spy, Xsnoop, Sendmail	Number of shell prompts invoked – the number of file creations	Buffer overflow
U2R	Buffer-overflow, Xterm, SQL attack, Perl, Loadmodule, Ps, Rootkit	Service reauested – connection duration – num of failed login attempts	Password guessing

Table 4.4: NSL-KDD attack types and classes [25]

Attack Class	**DoS**	**R2L**	**Prob**	**U2R**
Number	14920	5770	4842	134

Table 4.5: Number of different attack in NSL-KDD

In addition the numbers of normal connections are: 19422.

- **CIC-IDS2017**

The CICIDS2017 dataset contains the latest and mildest common attacks that look like real world data (PCAP). Also includes results of network traffic analysis using CICFlowMeter with data flows tagged based on timestamps, source and destination IP addresses, source and destination ports, protocols and attacks (CSV file). An extracted function definition is also available. Abstract behaviors of 25 users based on HTTP, HTTPS, FTP, SSH and Email were created for this dataset. The dataset extracted 84 network traffic features, with the last column being the multiclass category. In addition, this dataset meets 11 performance evaluation criteria compared to publicly available datasets from 1998 to 2016.

Chapter 4: Implementation, Result and Discussion

The following table 4.6 present the features of this dataset .

Num	Feature Name	Num	Feature Name	Num	Feature Name
1	Flow ID	29	Bwd IAT Max	57	Bwd Avg Bytes/Bulk
2	Source IP	30	Bwd IAT Min	58	Bwd Avg Packets/Bulk
3	Destination Port	31	Fwd PSH Flags	59	Bwd Avg Bulk Rate
4	Protocol	32	Bwd PSH Flags	60	Subflow Fwd Packets
5	Timestamp	33	Fwd URG Flags	61	Subflow Fwd Bytes
6	Flow Duration	34	Bwd URG Flags	62	Subflow Bwd Packets
7	Total Fwd Packets	35	Fwd Header Length	63	Subflow Bwd Bytes
8	Total Backward Packets	36	Bwd Header Length	64	Init_Win_bytes_forward
9	Total Length of Fwd Packets	37	Fwd Packets/s	65	Init_Win_bytes_backward
10	Fwd Packet Length Min	38	Bwd Packets/s	66	Act_data_pkt_fwd
11	Fwd Packet Length Max	39	Min Packet Length	67	Min_seg_size_forward
12	Fwd Packet Length Mean	40	Max Packet Length	68	Active Mean
13	Fwd Packet Length Std	41	Packet Length Mean	69	Active Std
14	Bwd Packet Length Max	42	Packet Length Std	70	Active Max
15	Bwd Packet Length Min	43	Packet Length Variance	71	Active Min
16	Bwd Packet Length Mean	44	FIN Flag Count	72	Idle Mena
17	Bwd Packet Length Std	45	SYN Flag Count	73	Idle Std
18	Flow Bytes/s	46	RST Flag Count	74	Idle Max
19	Flow Packets/s	47	PSH Flag Count	75	Idle Min
20	Flow IAT Mean	48	ACK Flag Count	76	Fwd Header Length
21	Flow IAT Std	49	URG Flag Count	77	Fwd Avg Bytes/Bulk
22	Flow IAT Max	50	CWE Flag Count	78	Fwd Avg Packets/Bulk
23	Flow IAT Min	51	ECE Flag Count	79	Fwd Avg Bulk Rate
24	Fwd IAT Total	52	Down/Up Ratio	80	Destination Port
25	Fwd IAT Mean	53	Bwd IAT Total	81	Total Length of Bwd Packets
26	Fwd IAT Std	54	Bwd IAT Min	82	Bwd IAT Mean
27	Fwd IAT Max	55	Bwd IAT Std	83	Average Bwd Segment Size
28	Fwd IAT Min	56	Average Packet Size	84	Average Fwd Segment Size

Table4.6:CICIDS2017 features [26]

The different attack classes in this dataset within the number of each one are presented in the following table:

Attack Class	DoS Hulk	DDoS	PortScan	DosGoldenEye
Number	172846	128016	90819	10286

Table 4.7: Number of different attack in CICIDS2017

In addition the numbers of BENIGN connections are: 2096134.

4.6 Test Protocol:

In our research, we developed a test protocol to do an investigation to prove its efficiency and quality that is shown in the following scheme :starting with data preparation moving to the most essential step which is Generation of pseudo images then CNN configuration , the training phase and finally the testing phase.

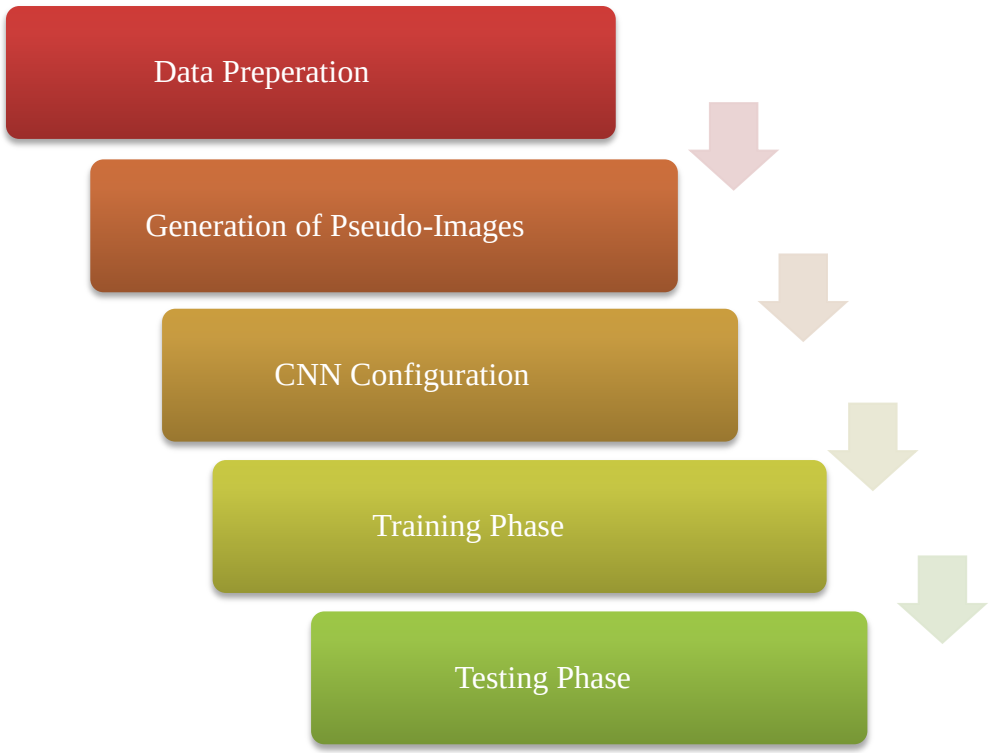


Figure 4.2: Protocol Scheme

Following scheme explains the different steps of the protocol starting from:

- Step one is for data preparation to make it able to fit in with the model to continue testing.
- Secondly which was the most important step is Generation of Pseudo-Images. The purpose was to focus on visualizing the aspect of images provided to CNN , while been presented in different sizes and frames.

- Third step was the CNN configuration to process inputs with the different shapes used, importing necessary libraries, adding dense layers.
- Then the training phase.
- Finally the testing phase to evaluate the model and then compare it.

4.6.1Data Preparation

The table 4.8 below defines the main steps to follow till both datasets be ready to use.

Data Preparation	
NSL-KDD	**CICIDS2017**
Remove class (difficulty level) Distribution of attack classes : (, DoS ,R2L, Prob and U2R based) Normalization One hot encoding categorical Encoding PIL Images	Remove class (difficulty level) Distribution of attack classes : ((BENIGN, DoS Hulk, PortScan, Dos GoldenEye) Normalization One hot encoding categorical Encoding PIL Images

Table 4.8: Data preparation (NSL-KDD, CICIDS2017)

After mentioning the main ideas to follow to get the data prepared, each idea will be explained in following:

- Remove attributes that has difficulty level:

This is the first step and it's important because when we remove all the attributes that have a difficult level the ones that we don't need in our case. We are going to gain memory usage .

- Redistribute across common attack class/ distribution of attack classes

The second step consists of looking for the correlation between the attributes , then breakdown of attack classes into normal, DoS ,R2L, Prob and U2R based on the correlation founded before.

- Normalization

In this step a data frame has created with multi-class labels that has already been mentioned in the previous step, and for normalizing a standard scaler has been applied.

- One hot encoding categorical

Chapter 4: Implementation, Result and Discussion

The dataset chosen contained attributes with string type and that needed to be converted to float type till we can convert the vectors into matrix. To achieve that a The ‘pd.get_dumnies ’ Function was used

For encoding label intrusion there are a little bit different for each dataset

For NSL-KDD: it contained 3 steps

- ✓ First creating a data frame with multi-class labels(Dos, Probe, R2L, U2R, normal)
- ✓ Second label encoding (0, 1, 2, 3, 4) multi-class labels(Dos, Probe, R2L, U2R, normal)
- ✓ Third, use one hot encoding categorical column to convert the attributes type string into float, then use drop .

For CICIDS2017: it contained 2 steps

- ✓ First creating a data frame with multi-class labels(BENIGN, DoS Hulk, PortScan, Dos GoldenEye)
- ✓ Second label encoding (0, 1, 2, 3, 4) multi-class labels(BENIGN, DoS Hulk, PortScan, Dos GoldenEye) .Then use drop.

4.6.2 Generation of Pseudo-Images:

The Generation of Pseudo-Images phase was the most important , were its principle was in the different ways used to present data while focusing on visualizing the aspect of images provided to CNN. While doing so with different images sizes(H×L) were put under test using different frames along with it, such as (32×32) frame , (64×64) or (128×128). In the table4.9 below some of the different image size that were used in testing are shown.

Test	Frames (HxL)
1	1x116
2	116x1
3	2x58
4	4x29
5	29x4
6	11x11
7	10x13
8	13x10

Table 4.9: Different image sizes

- **Figure 4.3** presented two images of (32x32) with two different sizes (4 x29), (29 x4) in a manipulative way in (H x L) .While **Figure 4.4** presented the same idea in (64x64) frame and **figure 4.5** in (128x128) frame.

Chapter 4: Implementation, Result and Discussion

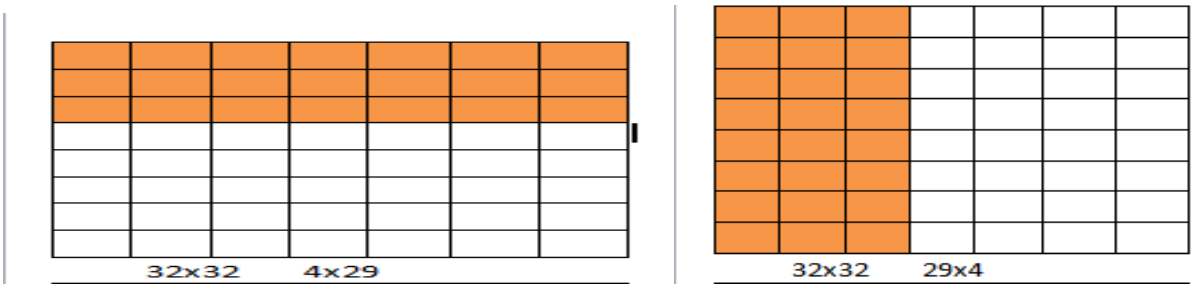


Figure 4.3: Image frame (32x32)

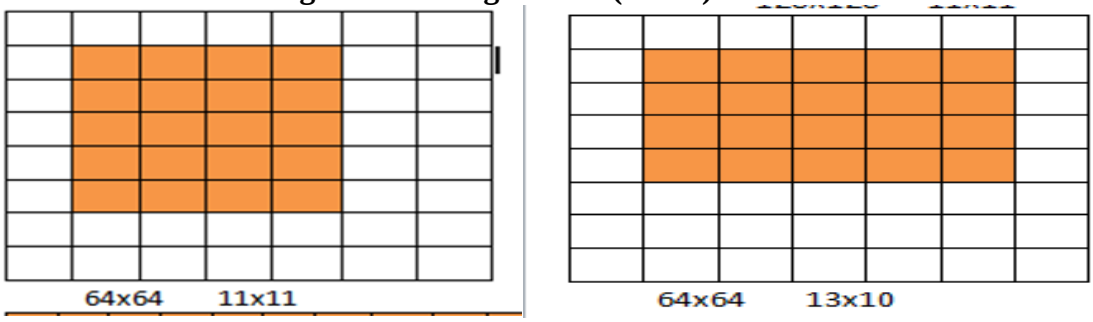


Figure 4.4: Image frame (64x64)

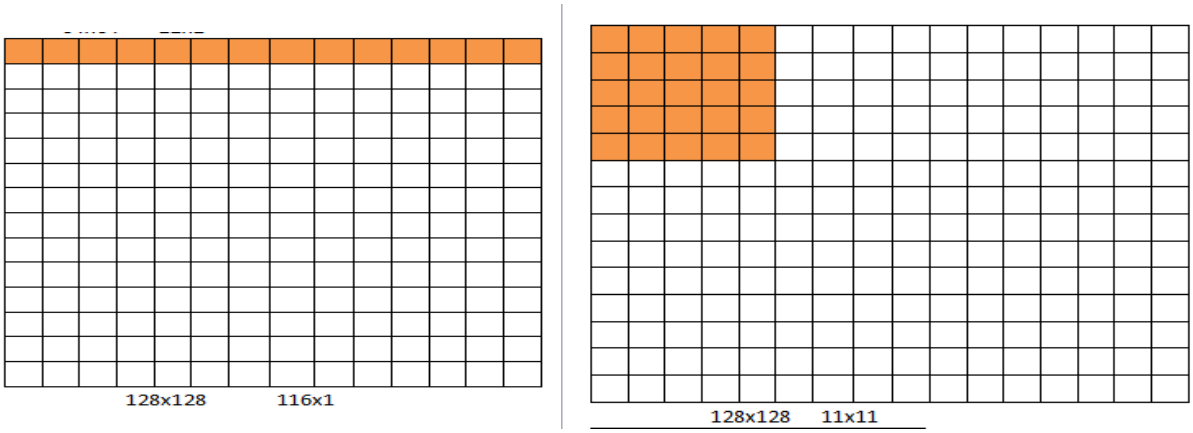


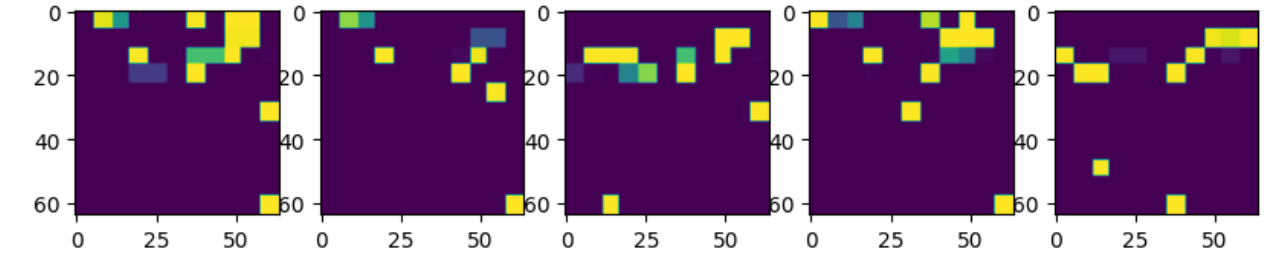
Figure 4.5: Image frame (128x128)

- **Resizing Image:**

In our research, to prove the efficiency of our model we needed a test protocol which means resizing every time the images frame starting with (32×32) to (64×64) till (128×128), before that we convert vectors into matrix size (H×L) . It was important for the size images to be compatible with the inputs of the learning model, they are resized into a frame of (32x32), (64x64) or (128x128).This cropping can be done with or without resampling. When resampling is enabled, the image is resized with a resample method, while when this option is disabled, the image is simply cropped. We defined it simply in our code by **Resmap = true or Resamp = False** in the part of image generation, and it affects the results for sure.

Chapter 4: Implementation, Result and Discussion

Here are three examples of the images we got with a frame (64×64), (128×128) size (11×11) with resampling (true and false). Plus, when resamp = false it's more clear than when it's true, frame (128×128) proves that it's better than (32×32) or (64×64), such as in (1×116) case there is no such a noticeable difference. Some of different matrix seizes that we worked on are described below in the following images:



.

Figure 4.6: Test images of (11×11) frame 64 resamp= false

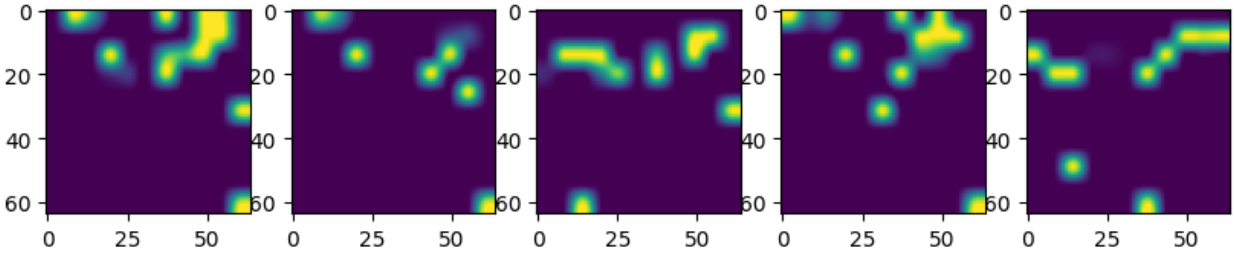


Figure 4.7: Test images of (11×11) frame 64 resamp= true

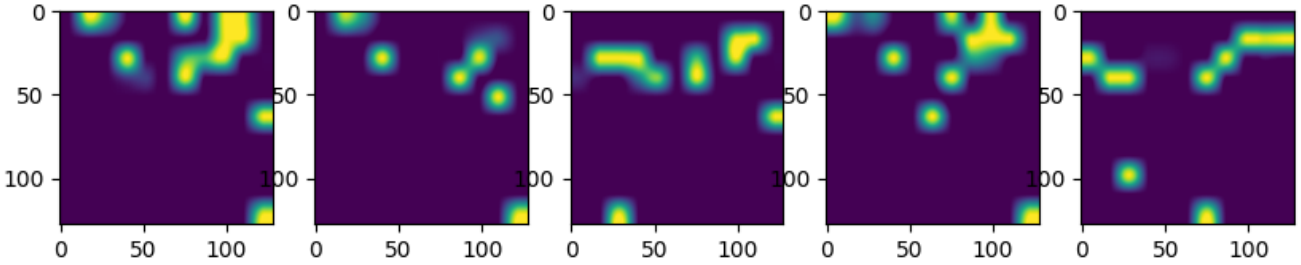


Figure 4.8: Test images of (11×11) frame (128×128) resamp= true

- **training phase**

Convert the vectors into matrix then images and Divide data into training and testing sets, a *make asymmetric image function was operated while tests started from 20%*

- **test phase**

Different metrics were applied to evaluate the model : accuracy, precision, recall, f1score and execution time

Chapter 4: Implementation, Result and Discussion

4.7 Result and Discussion

Multiple experiments were conducted using different datasets, training sizes, classifiers, and image sizes to ensure that the results obtained aligned with our hypotheses. To achieve this, several testing phases were carried out, following the following sequence:

1. The first test involved examining the impact of varying image sizes with different frames for each dataset. This test aimed to identify any differences that may arise from these variations.

2. In the second test, we selected a reference model based on the outcomes of the first test. We then compared the results obtained from each dataset using this reference model.
3. The third test focused on evaluating the effect of changing the learning dataset size on both datasets. By manipulating the learning dataset size, we aimed to compare the performance and outcomes.
4. The subsequent test involved comparing our CNN model to the Random Forest classifier. This test aimed to discern any significant variations in performance and outcomes between these two models.

By systematically conducting these experiments and tests, we ensured that the results obtained would provide reliable and confident evidence to support our hypotheses.

4.7.1 - Test N°1:

The idea of the first test was to use different size images and frames as well with the both datasets. We used 80% of the dataset for training the CNN and the other 20% were used for testing. With specific parameters of configuration as Table 4.10 shown below.

Epoch number	Batch size	Random State	Optimizers
10	32	42	Adam

Table4.10: The parameters selected for our model.

- ✓ **Codification for the tests :**

We used a codification for the presentation of different combinations as mentioned in table 4.11 below

Codification	signification
11× 11/64/true	**11x11** is the frame size **64** is the image size **True** for resampling.

Table4.11: Codification with signification

Chapter 4: Implementation, Result and Discussion

Part 1 : NSL-KDD Results: The following tables present the results of our tests. Table 4.12 and table 4.13 are presenting the results for NSL-KDD dataset, when testing with reasmp = false are shown in the first table and testing with reasmp = true shown in the second table.

Test	Description	Accuracy	Precision	Recall	F1-Score	Temps d'exécutior	L	H	LxH	Taille
1	4x29/32/false	0,9900	0,9899	0,9900	0,9899	2,9628	4	29	116	38KB
2	29x4/32/false	0,9906	0,9905	0,9906	0,9904	3,1471	29	4	116	38KB
3	11x11/32/false	0,9890	0,9888	0,9890	0,9889	2,6604	11	11	121	38KB
4	10x13/32/false	0,9903	0,9902	0,9903	0,9901	2,0604	13	10	130	38KB
5	2x58/64/false	0,9897	0,9896	0,9897	0,9894	2,4309	2	58	116	42KB
6	4x29/64/false	0,9895	0,9894	0,9895	0,9894	3,0174	4	29	116	42KB
7	29x4/64/false	0,9916	0,9915	0,9916	0,9915	2,4903	29	4	116	43KB
8	11x11/64/false	0,9915	0,9915	0,9915	0,9915	2,9754	11	11	121	43KB
9	10x13/64/false	0,9903	0,9901	0,9903	0,9901	2,8305	10	13	130	43KB
10	1x116/128/false	0,9858	0,9856	0,9858	0,9856	5,9638	1	116	116	39KB
11	116x1/128/false	0,9859	0,9855	0,9859	0,9855	5,4196	116	1	116	39KB
12	2x58/128/false	0,9902	0,9901	0,9902	0,9901	5,9878	2	58	116	39KB
13	4x29/128/false	0,9908	0,9906	0,9908	0,9906	6,0430	4	29	116	40KB
14	29x4/128/false	0,9920	0,9919	0,9920	0,9919	6,3426	29	4	116	41KB
15	11x11/128/false	0,9912	0,9911	0,9912	0,9911	6,0402	11	11	121	41KB
16	10x13/128/false	0,9903	0,9901	0,9903	0,9901	5,8650	10	13	130	41KB

Table 4.12: NSL-KDD Test Results Frame(32×32,64×64,128×128) resamp= false

Test	Description	Accuracy	Precision	Recall	F1-Score	Temps d'exécution	L	H	LxH	Taille
1	4x29/32/true	0,9862	0,9861	0,9862	0,9857	2,9628	4	29	116	45KB
2	29x4/32/true	0,9892	0,9890	0,9892	0,9890	2,2357	29	4	116	44KB
3	11x11/32/true	0,9916	0,9914	0,9916	0,9915	2,1179	11	11	121	45KB
4	10x13/32/true	0,9908	0,9907	0,9908	0,9907	2,7203	13	10	130	44KB
5	13x10/32/true	0,9918	0,9916	0,9918	0,9916	2,6719	10	13	130	45KB
6	2x58/64/true	0,9879	0,9877	0,9879	0,9878	2,5114	2	58	116	70KB
7	4x29/64/true	0,9873	0,9871	0,9873	0,9872	2,8610	4	29	116	80KB
6	29x4/64/true	0,9895	0,9895	0,9895	0,9894	2,7273	29	4	116	76KB
9	11x11/64/true	0,9924	0,9922	0,9924	0,9922	2,9495	11	11	121	87KB
10	10x13/64/true	0,9918	0,9917	0,9918	0,9917	2,5356	10	13	130	83KB
11	13x10/64/true	0,9921	0,9920	0,9921	0,9919	2,9347	13	10	130	79KB
12	1x116/128/true	0,9855	0,9854	0,9855	0,9853	5,9335	1	116	116	39KB
13	116x1/128/true	0,9825	0,9821	0,9825	0,9822	4,9173	116	1	116	40KB
14	2x58/128/true	0,9906	0,9904	0,9906	0,9904	6,0790	2	58	116	70KB
15	4x29/128/true	0,9908	0,9906	0,9908	0,9906	6,2801	4	29	116	78KB
16	29x4/128/true	0,9935	0,9934	0,9935	0,9934	6,3320	29	4	116	74KB
17	11x11/128/true	0,9936	0,9935	0,9936	0,9935	5,2314	11	11	121	86KB
18	10x13/128/true	0,9928	0,9927	0,9928	0,9928	5,2204	10	13	130	83KB
19	13x10/128/true	0,9933	0,9934	0,9933	0,9933	5,3668	13	10	130	76KB

Table 4.13: NSL-KDD Test Results Frame(32×32,64×64,128×128) resamp= true

Chapter 4: Implementation, Result and Discussion

For the NSL-KDD the Figure 4.9 presents a precision graphic and figure 4.10 present the execution time. The first comparison is with Resamp = False and for all different frames and images size.

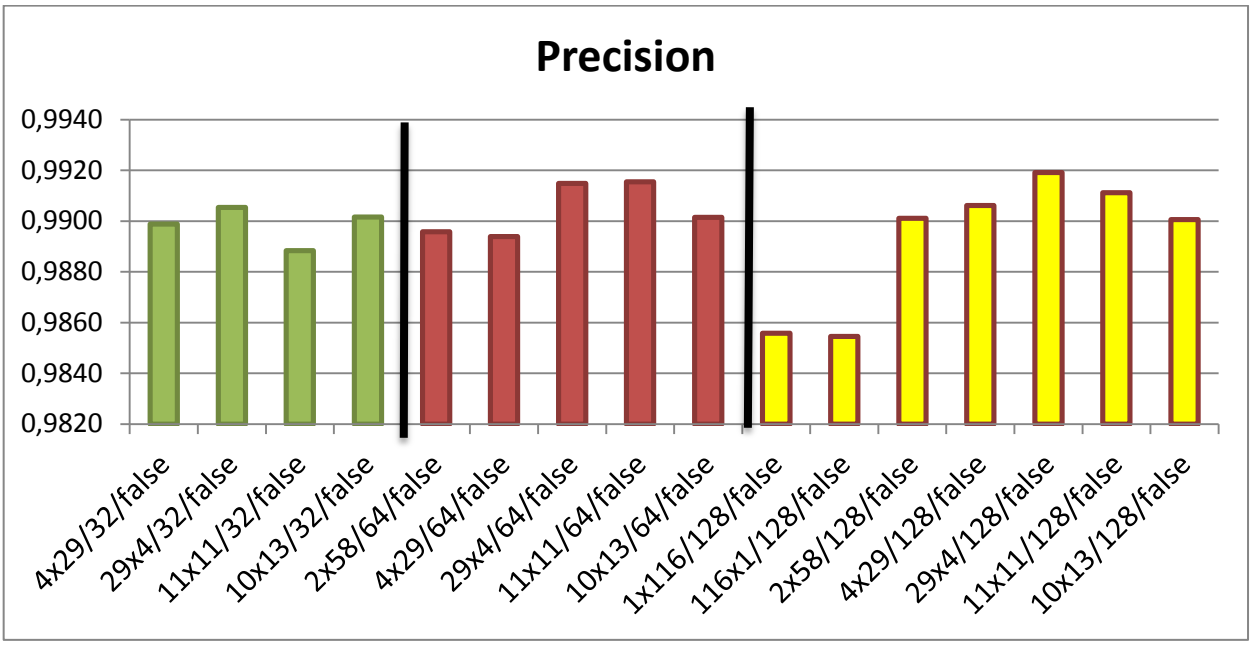


Figure 4.9: Precision Graphic Results for NSL-KDD dataset resamp= false

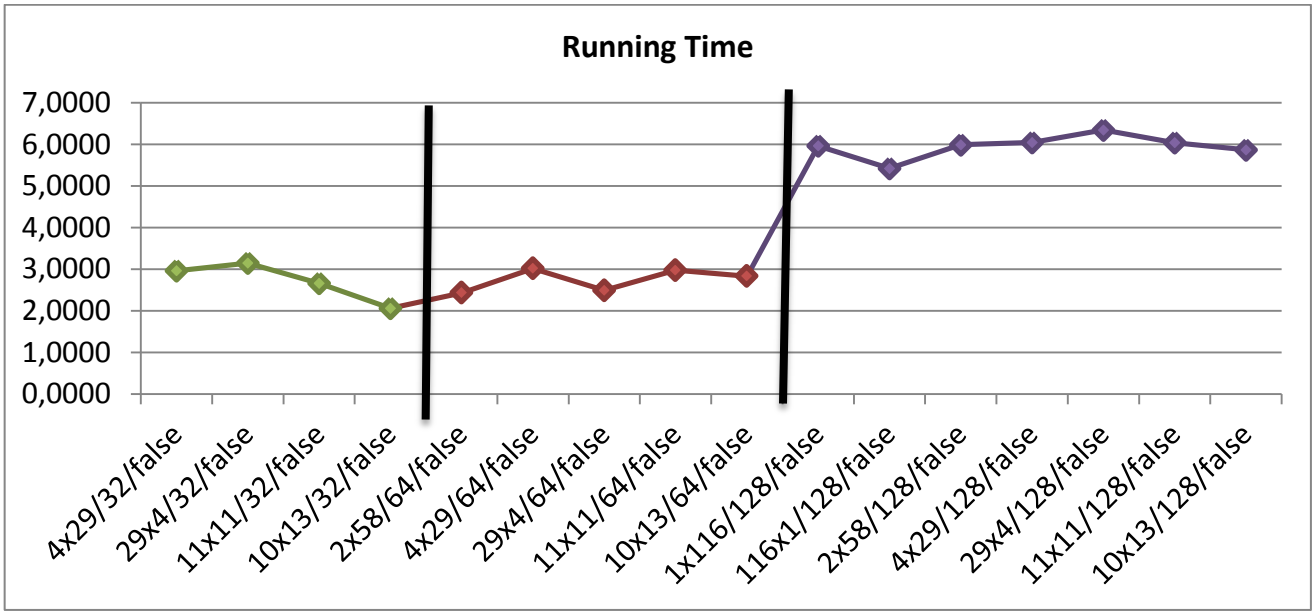


Figure 4.10: Running Time Graphic Results for NSL-KDD dataset resamp= false

For the NSL-KDD the Figure 4.11 presents a precision graphic and figure 4.12 present the execution time. The second comparison is with Resamp = True and for all different frames and images size.

Chapter 4: Implementation, Result and Discussion

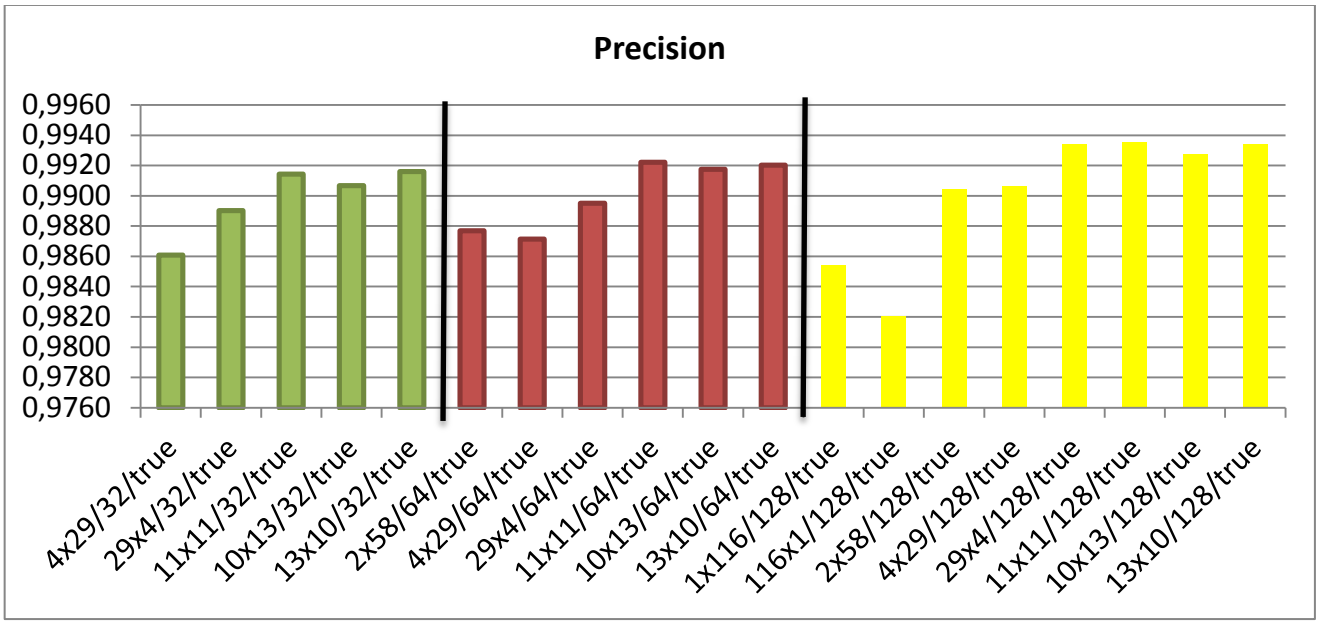


Figure 4.11: Precision Graphic Results for NSL-KDD dataset resamp= true

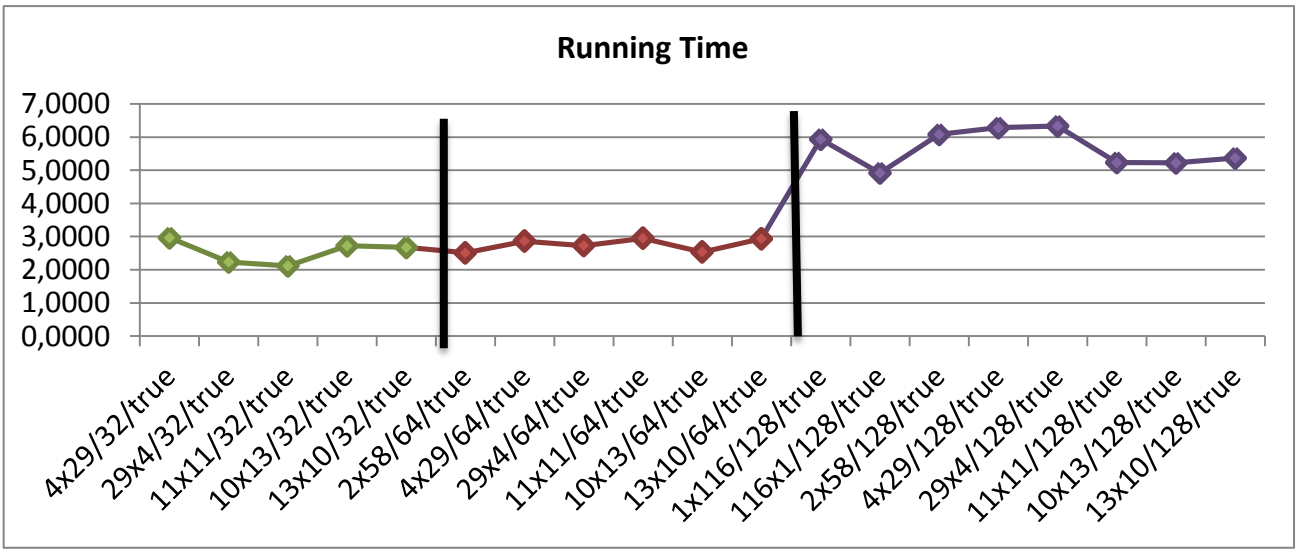


Figure 4.12: Running Time Graphic Results for NSL-KDD dataset resamp= true

Chapter 4: Implementation, Result and Discussion

Part 2: CICIDS2017 Results: The following tables present the results of our tests. Table 4.14 and table 4.15 are presenting the results for CICIDS2017 dataset, when testing with reasmp = false are shown in the first table and testing with reasmp = true shown in the second table

Test	Description	Accuracy	Precision	Recall	F1-Score	Temps d'exécution	L	H	LxH	Taille
1	4x29/32/false	0,9863	0,9863	0,9863	0,9863	0,8951	4	29	116	56Ko
2	29x4/32/false	0,9876	0,9875	0,9876	0,9875	1,1247	29	4	116	56Ko
3	11x11/32/false	0,9918	0,9919	0,9918	0,9918	1,3738	11	11	121	56Ko
4	10x13/32/false	0,9894	0,9894	0,9894	0,9894	1,3776	13	10	130	56Ko
5	2x58/64/false	0,9921	0,9922	0,9921	0,9921	1,4913	2	58	116	59Ko
9	4x29/64/false	0,9889	0,9889	0,9889	0,9889	1,3961	4	29	116	59Ko
7	29x4/64/false	0,9916	0,9916	0,9916	0,9916	0,8916	29	4	116	60Ko
8	11x11/64/false	0,9915	0,9916	0,9915	0,9915	1,9166	11	11	121	60Ko
9	10x13/64/false	0,9906	0,9906	0,9906	0,9906	1,4314	10	13	130	60Ko
10	1x116/128/false	0,9820	0,9821	0,9820	0,9820	3,1600	1	###	116	56Ko
11	116x1/128/false	0,9837	0,9838	0,9837	0,9836	3,1548	###	1	116	57Ko
12	2x58/128/false	0,9880	0,9880	0,9880	0,9880	2,8024	2	58	116	57Ko
13	4x29/128/false	0,9876	0,9878	0,9876	0,9876	2,6805	4	29	116	57Ko
14	29x4/128/false	0,9902	0,9902	0,9902	0,9902	2,7949	29	4	116	57Ko
15	11x11/128/false	0,9911	0,9911	0,9911	0,9911	2,9929	11	11	121	58Ko
16	10x13/128/false	0,9928	0,9928	0,9928	0,9928	2,9268	10	13	130	58Ko

Table 4.14: CICIDS2017 Test Results Frame (32×32,64×64,128×128) resamp= false

Test	Description	Accuracy	Precision	Recall	F1-Score	Temps d'exécution	L	H	LxH	Taille
1	4x29/32/true	0,9911	0,9911	0,9911	0,9911	1,3691	4	29	116	62Ko
2	29x4/32/true	0,9900	0,9900	0,9900	0,9900	0,7603	29	4	116	62Ko
3	11x11/32/true	0,9919	0,9919	0,9919	0,9919	0,7767	11	11	121	63Ko
4	10x13/32/true	0,9895	0,9895	0,9895	0,9895	0,9464	13	10	130	62Ko
5	13x10/32/true	0,9875	0,9875	0,9875	0,9875	1,4242	10	13	130	61Ko
6	2x58/64/true	0,9916	0,9916	0,9916	0,9916	0,8928	2	58	116	89Ko
7	4x29/64/true	0,9948	0,9949	0,9948	0,9948	0,8836	4	29	116	94Ko
8	29x4/64/true	0,9924	0,9924	0,9924	0,9924	1,4865	29	4	116	92Ko
9	11x11/64/true	0,9936	0,9936	0,9936	0,9936	0,8824	11	11	121	99Ko
10	10x13/64/true	0,9941	0,9941	0,9941	0,9941	1,4304	10	13	130	96Ko
11	13x10/64/true	0,9872	0,9873	0,9872	0,9872	1,4408	13	10	130	95Ko
12	1x116/128/true	0,9842	0,9842	0,9842	0,9841	3,0594	1	###	116	57Ko
13	116x1/128/true	0,9859	0,9860	0,9859	0,9859	2,9207	###	1	116	58Ko
14	2x58/128/true	0,9947	0,9947	0,9947	0,9947	3,1856	2	58	116	88Ko
15	4x29/128/true	0,9962	0,9962	0,9962	0,9962	3,0076	4	29	116	92Ko
16	29x4/128/true	0,9948	0,9948	0,9948	0,9948	2,9173	29	4	116	90Ko
17	11x11/128/true	0,9928	0,9928	0,9928	0,9928	2,6567	11	11	121	99Ko
18	10x13/128/true	0,9954	0,9954	0,9954	0,9954	3,0371	10	13	130	95Ko
19	13x10/128/true	0,9948	0,9948	0,9948	0,9947	2,9284	13	10	130	94Ko

Table 4.15: CICIDS2017 Test Results Frame (32×32,64×64,128×128) resamp= true

Chapter 4: Implementation, Result and Discussion

For the CICIDS2017 the Figure 4.13 presents a precision graphic and figure 4.14 present the execution time. The third comparison is with Resamp = False and for all different frames and images size.

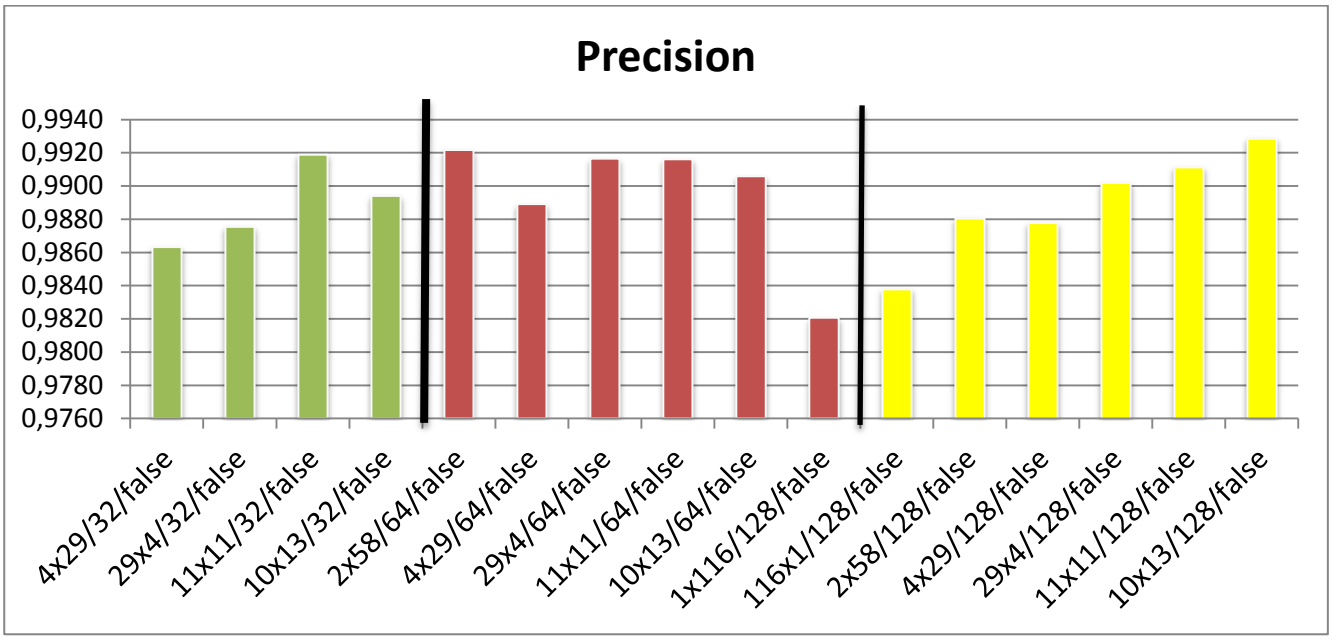


Figure 4.13: Precision Graphic Results for CICIDS2017 dataset resamp = false

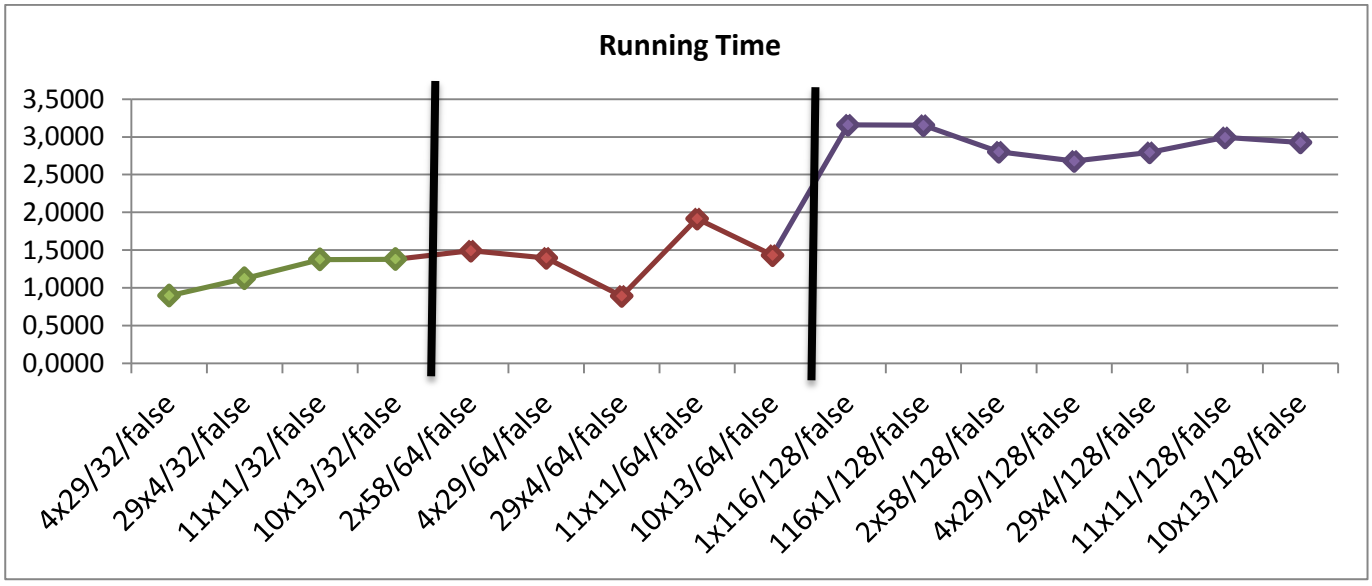


Figure 4.14: Running Time Graphic Results for CICIDS2017 dataset resamp = false

Chapter 4: Implementation, Result and Discussion

For the CICIDS2017 the Figure 4.15 presents a precision graphic and figure 4.16 present the execution time. The forth comparison is with Resamp = True and for all different frames and images size.

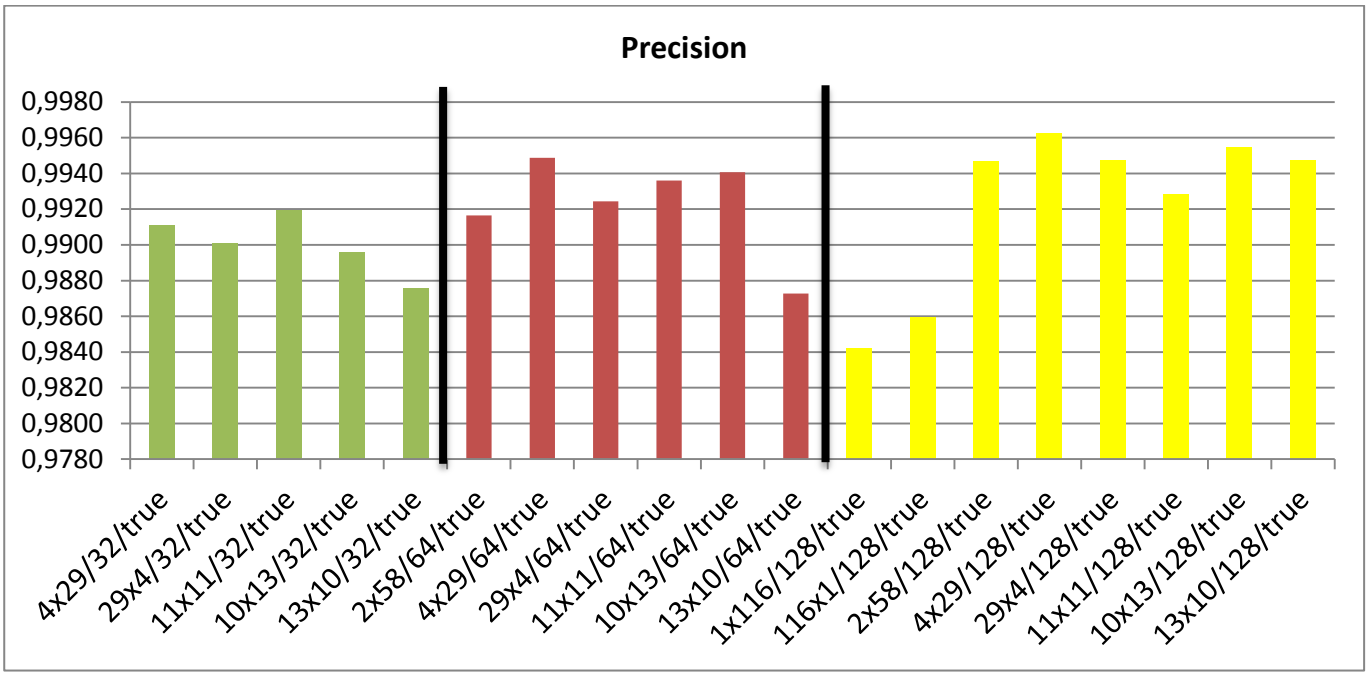


Figure 4.15: Precision Graphic Results for CICIDS2017 dataset resamp = true

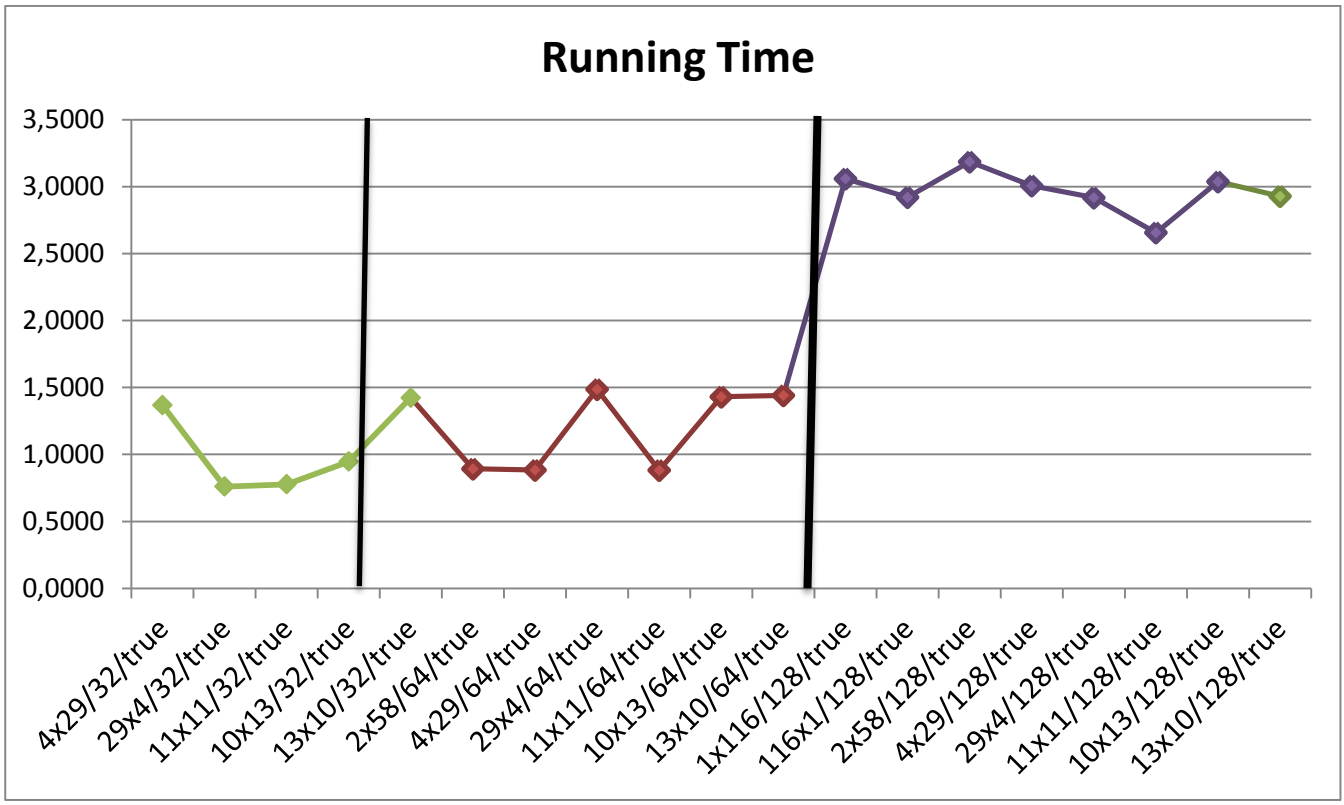


Figure 4.16: Running Time Graphic Results for CICIDS2017 dataset resamp = true

Chapter 4: Implementation, Result and Discussion

Now from what is observed from the above tables and graphics for the both datasets, here are some points extracted :

1. Best choice is (11×11/64/true) chosen because of the quality in what the precision is the highest as well as because the timing is better and optimal as well as the plus is that this choice is square we notice that (4×29) 's quality is good and presentable as well.
2. while for (1× 116) isn't advisable at all as shown in the figures and that's because of the choice is a flat one
3. What is noticed from the numbers and graphics is that the (128×128) frame is with good quality results but the timing is that high as shown in the above figures. Which is not recommended because it paralyzes the IDS work to figure out a response in an ideal timing for the system
4. Sampling True provided us with better results than false one.
5. From what is noticed that when sampling is False the size is much smaller than when it is true
6. Ids size depends particularly on the size of the frame of representation of the data and not on the size of the images
7. The choice for best configuration to balance between precession and running time was on 64 frames which offered an impressive precession and an optimal running time.
8. the results in the graphics above makes (11× 11/64/true) a best case combination for our tests. When **11x11** is the frame size and **64** is the image size and **true** for resampling.

4.7.2 - Test N°2:

Part one:

In this test we took a reference model from the first test (11x1164/true) , and compared the result of each dataset as codification precision figure 4.17 and codification running time figure 4.18 with both resampling = true and false shows .

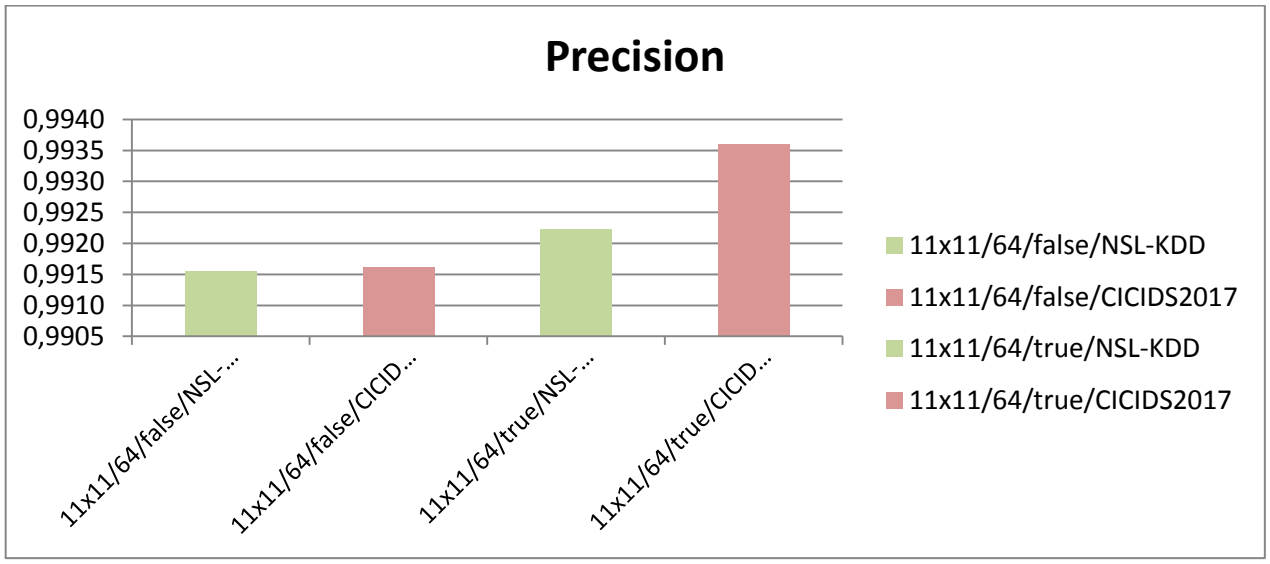


Figure 4.17: Codification precision Graphic Results for both datasets resamp = true/false

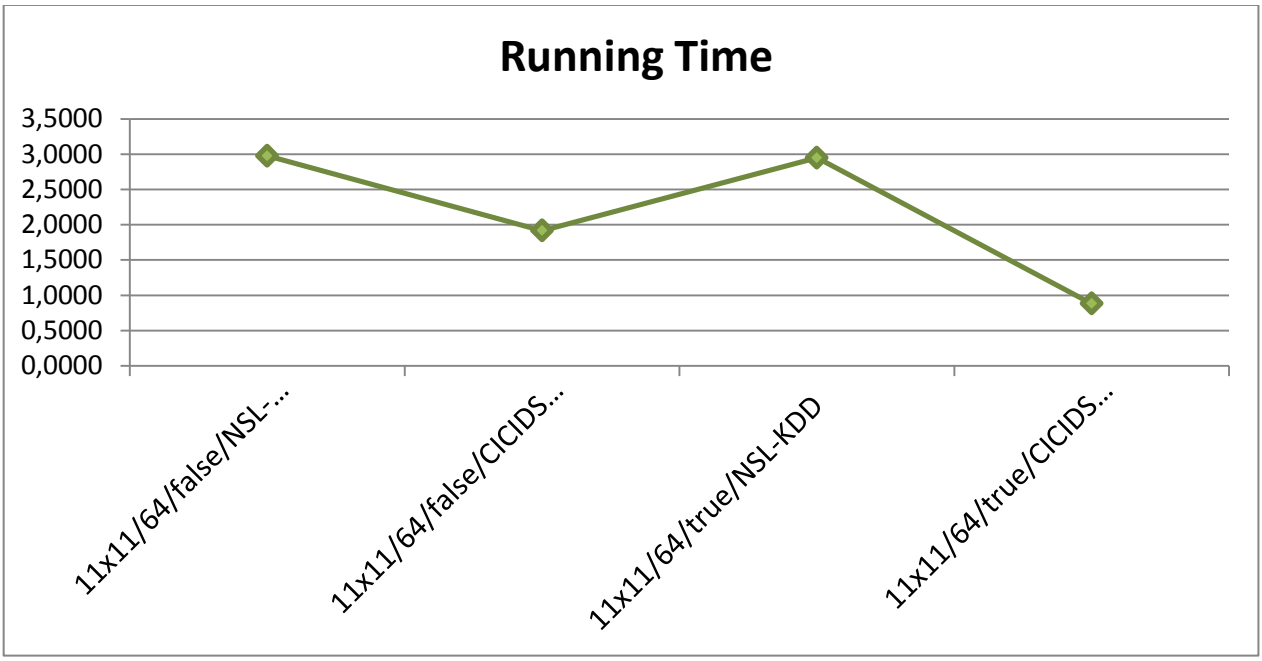


Figure 4.18: Codification running Time Graphic Results for both datasets resamp = true/false

Chapter 4: Implementation, Result and Discussion

From what is noticed from figures 4.17 and 4.18 above that codification(11x11/64/true) gives better precision and less running time in CICIDS2017 dataset with resamp = true then the one given for NSL-KDD dataset.

Part two: Learning curves

Figures 4.19 and 4.20 presents the Learning curves for both datasets NSL-KDD, CICIDS2017 used, as shown below our CNN model is with an ideal accuracy for training while the loss is approximately in a low range. Which proves that our CNN model is with high training capacity .

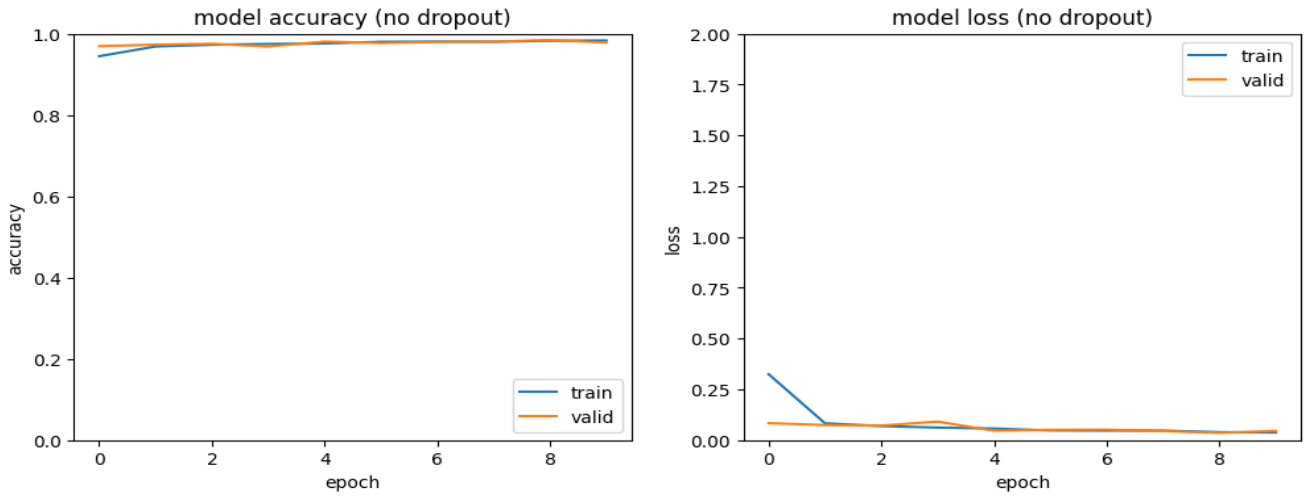


Figure 4.19: the accuracy and the loss function for NSL-KDD.

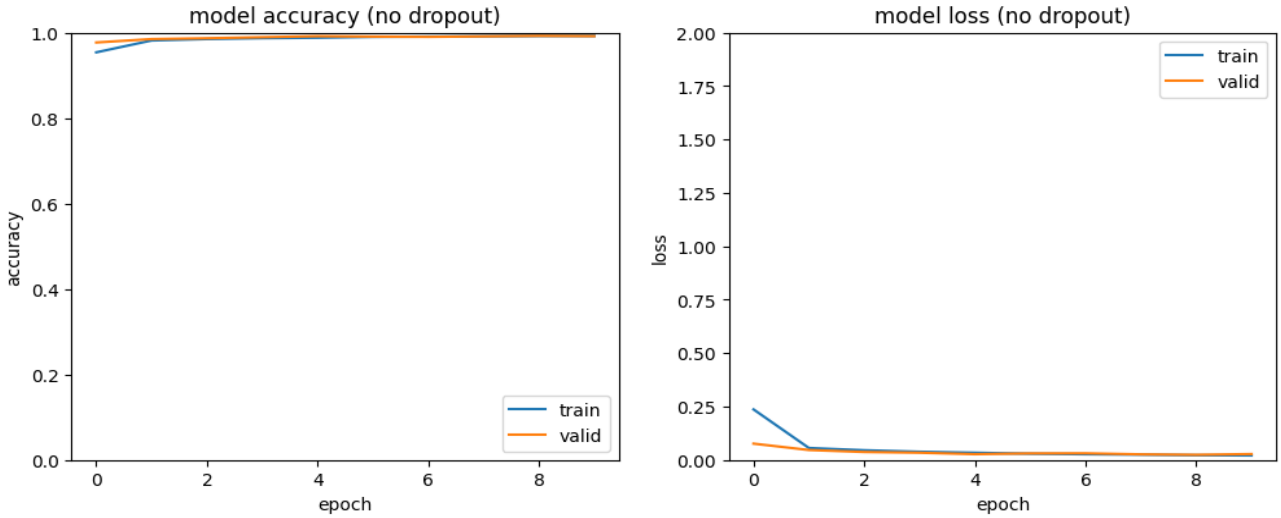


Figure 4.20: the accuracy and the loss function for CICIDS2017.

4.7.3 - Test N°3:

The purpose of the third test is how the Learning dataset size affect the quality of the CNN prediction? We choose the best codification of CNN from the first test. Which is (11x11x64xtrue)? For doing the test we used different size from 80% to 20% on both datasets. Where we used 80% of the dataset for training, while the rest 20% were used for testing, and other sizes such as 70% of the dataset used for training while the rest 30% is used for testing.

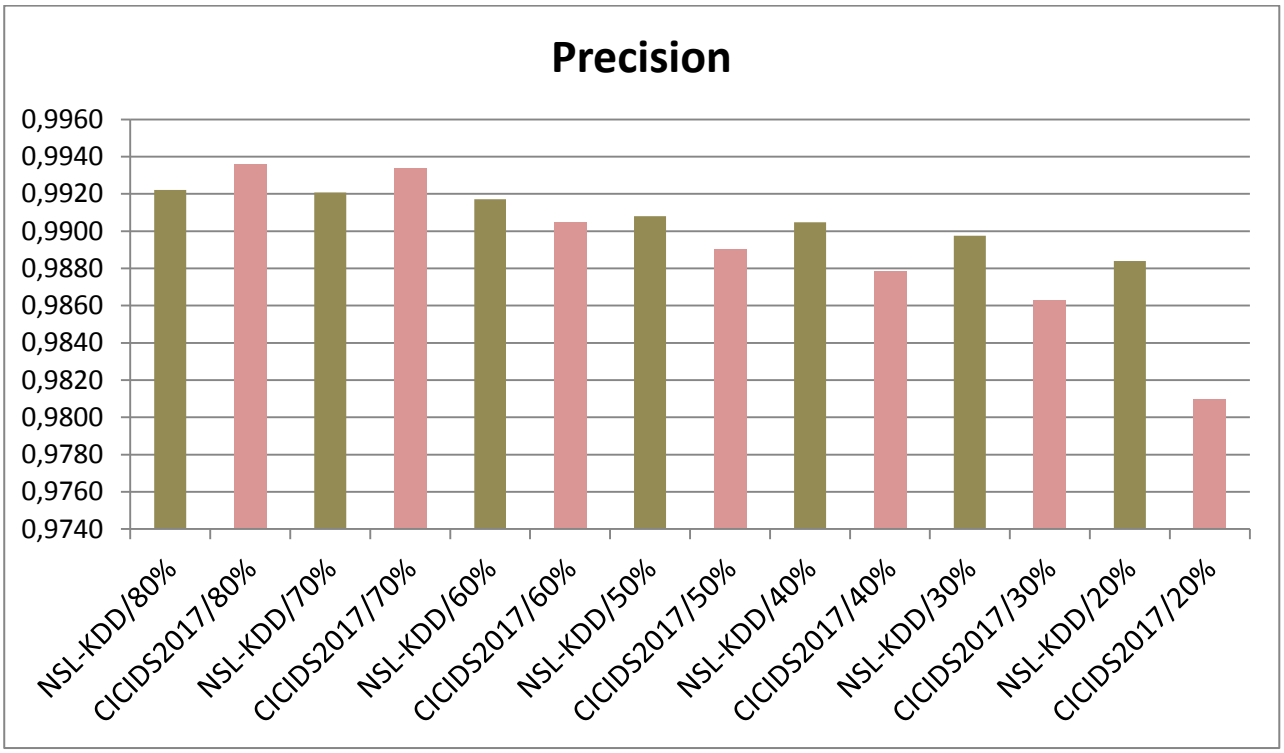


Figure 4.21: Precision Comparison Graphic Results for both datasets

From what was declared previously the choice for best configuration to balance between precession and running time was on 64 frames which offered an impressive precession and an optimal running time. From the figure 4.21 we came to conclude two principal points:

1. CNN quality is smashing despite the fact that training is 20% while precession is at 0.9880 as shown in figure 4.21.
2. Next remark is that the dataset type influence the quality as it is previewed in the figure 4.21 above, with the fact that precession in NSL-KDD went from 0.9920 to 0.9880 and kept in this level while in CICIDS2017 case the precession went from 0.9940 which is the highest to 0.9810 the lowest that itself proves the difference that dataset can make that affects the quality.

4.7.4 - Test N°4:

In order to compare the results of our model another classifier was chosen to do so. Which was Random forest, pointing out the differences between the two?

- First of all, Random Forests (RF) and Convolutional Neural Networks (CNN) are different types of algorithms.
- RF is a set of decision trees. Each decision tree in the assembly processes a sample and expects an output label (if classified). Team decision trees are independent. Anyone can predict the final answer.
- A convolutional neural network is the class of artificial neural networks most commonly used to analyze visual images. CNNs use a mathematical operation called convolution instead of general.

- Matrix multiplication on at least one of their levels. They are specially designed to process pixel data and are used for image recognition and processing.
- Random forests can only work with tabular data. (tabular data is data in a table format) On the other hand, a convolutional neural network can work with many different types of data: tabular data, images, audio data.

The following graphic in figure 4.22 presents testing result (precision) for a comparison between CNN and RF using NSLKDD dataset and choosing codification (11x11x64xtrue) with different learning size (from 80% till 20%)

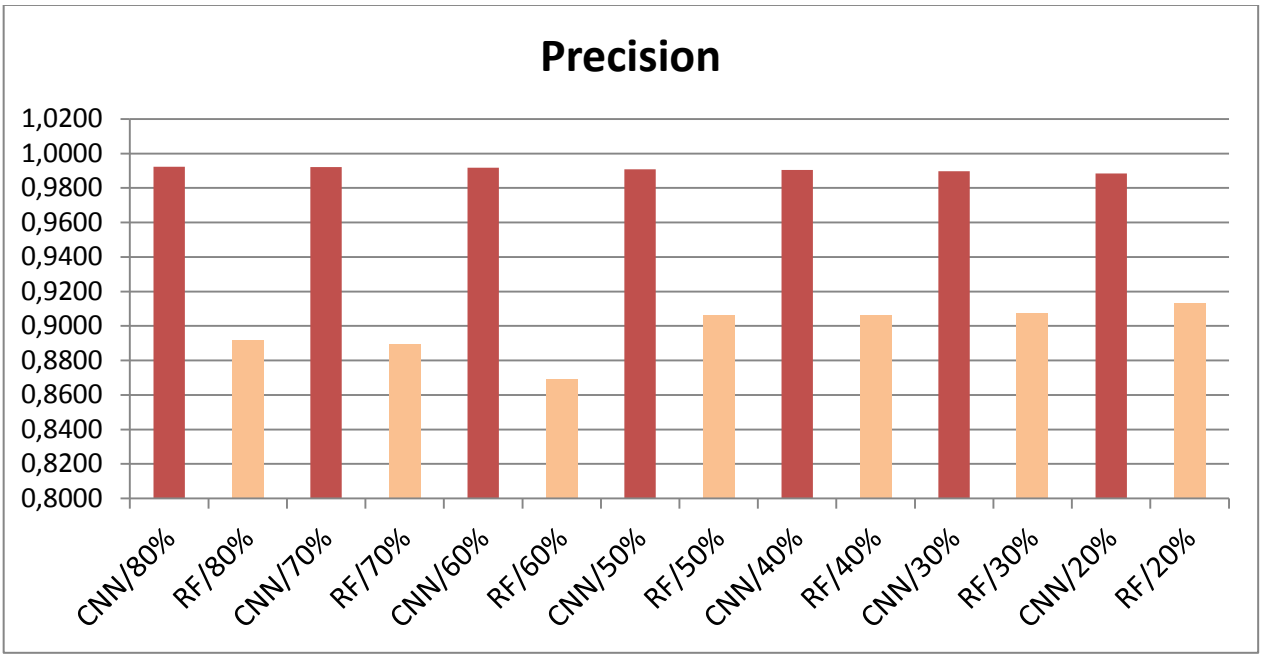


Figure 4.22: comparison between CNN and RF using NSLKDD dataset

The following graphic in figure4.23 presents testing result (precision) for a comparison between CNN and RF using CICIDS2017 dataset and choosing codification (11x11x64xtrue) with different learning size (from 80% till 20%)

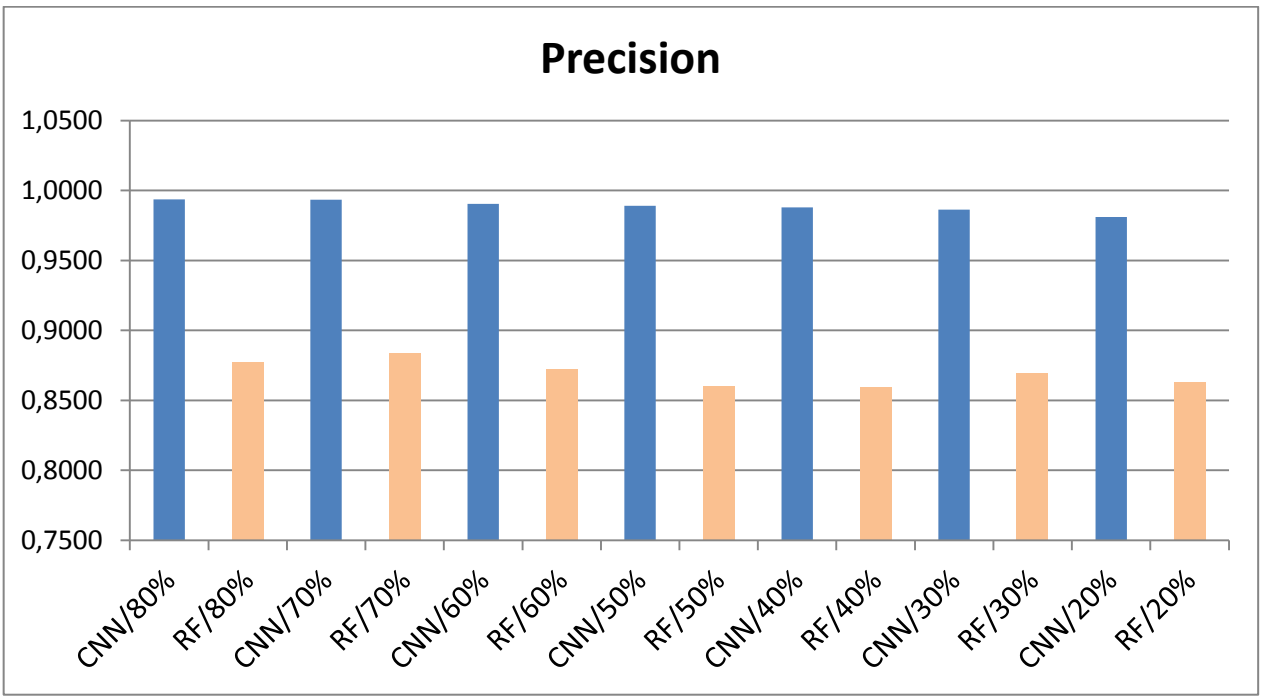


Figure 4.23: comparison between CNN and RF using CICIDS2017 dataset

- **Comparison of CNN and RF using confusion matrix:**

  → We used confusion matrix with different learning size and we took two examples , the 80% and the 20% for comparing our model CNN with the other classifier chosen the RF, the following figures 4.24 and 4.25 presents the matrix of this comparison using NSL-KDD dataset.

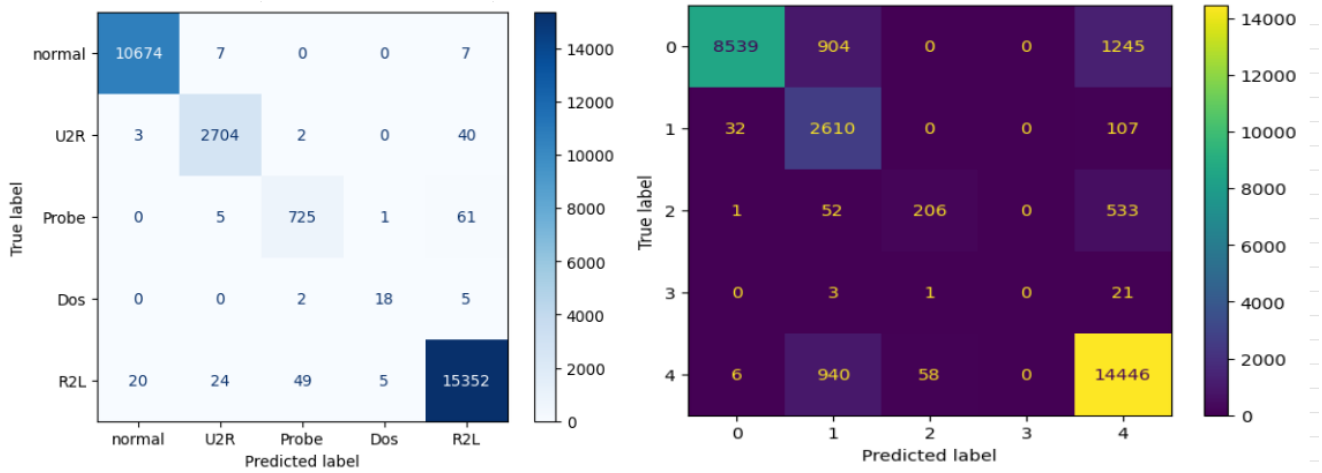


Figure 4.24: Comparison using confusion metrics with 80% of learning size (in the Left CNN results, in Right RF results) NSL-KDD

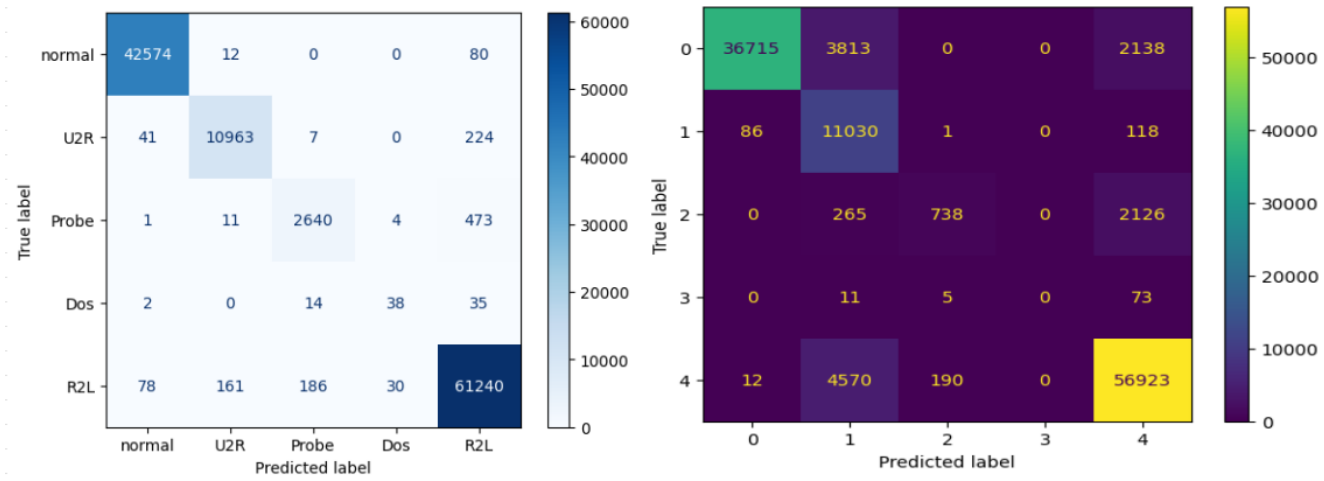


Figure 4.25: Comparison using confusion metrics with 20% of learning size (in the Left CNN results, in Right RF results) NSL-KDD

In this test, we compared the predicted label of CNN and RF as shown in figures 4.24 and 4.25 above using NSL-KDD dataset .Based on that, the prediction with 80% learning size or 20% for CNN was the best ,we notice that in class 3: for R2L example with 80% of learning size, CNN predicted

15352 correct with 0.006% predicted as a false classification (FC) ,while RF results for R2L were 14446 predicted correct and 0.07% were predicted false , for U2R CNN predicted TP= 2704 and 0.016% as false classification and . Now for 20% of learning size the performance of CNN kept an ideal level were U2R Eye prediction gave us TP = 2870 and 0.09 % as false classification , in contrary RF could predict less TP = 738 while 3.23% as false classification .So the confusion matrix above confirms the idea that CNN predicts much better than RF in this case.

→ For comparing our model CNN with RF using CICIDS2017 dataset with different learning size the following figure 4.26 and figure 4.27 presents the matrix of this comparison.

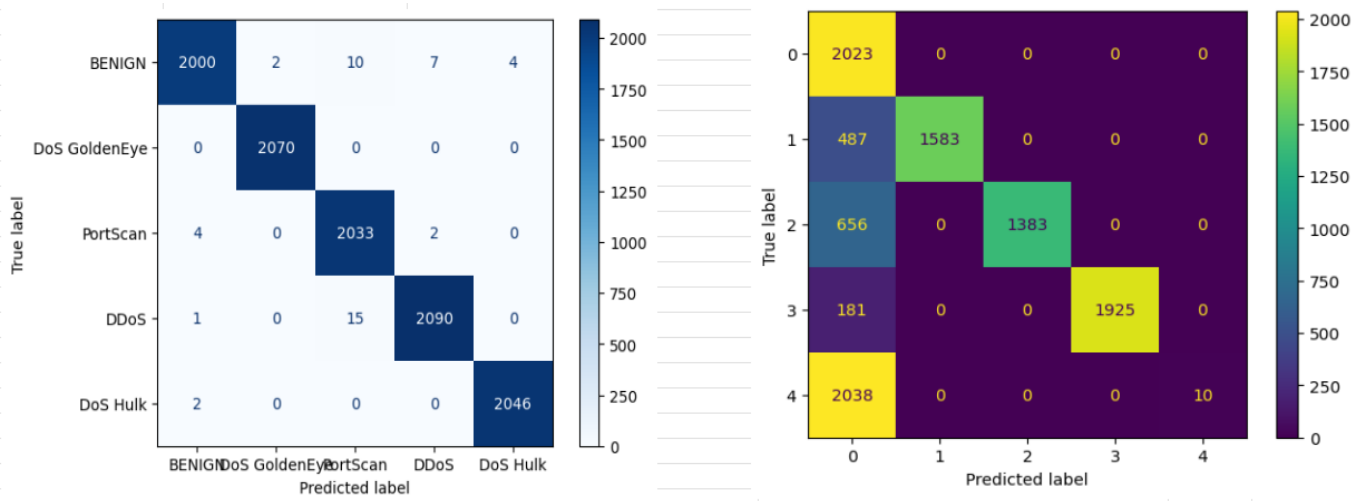


Figure 4.26: Comparison using confusion metrics with 80% of learning size (in the Left CNN results, in Right RF results) CICIDS2017

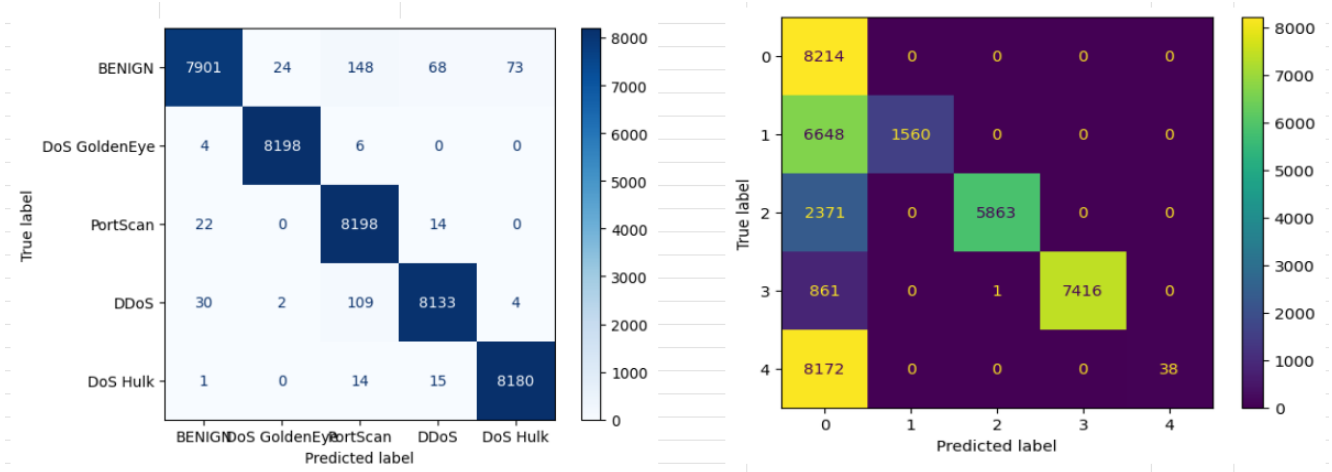


Figure 4.27: Comparison using confusion metrics with 20% of learning size (in the Left CNN results, in Right RF results) CICIDS2017

Chapter 4: Implementation, Result and Discussion

In this section, we conducted a comparison between the predicted labels of CNN and RF models using CICIDS2017 dataset, as depicted in Figure 4.26 and Figure 4.27 Based on this analysis, we observed the following results:

For the "DoS Golden Eye" attack, CNN correctly predicted all 2070 instances as true positives, whereas RF predicted 1583 instances as true positives and had 487 instances classified as false negatives. In the case of the "DDoS" example, CNN achieved 2090 true positives with only 0.008% false classifications, while RF predicted 1925 true positives with 0.094% false negatives.

When considering a learning size of 20%, the performance of CNN remained at an ideal level. For the "DoS Golden Eye" attack, CNN correctly predicted 8198 true positives with only 0.004% false classifications. On the other hand, RF predicted a lower number of true positives with only 1560 instances and a higher false negative rate of 4.26%.

Based on these results, we concluded that CNN was a significantly better predictor. Additionally, we observed that RF had a higher number of false positives compared to true positives. Moreover, the comparison of the two models using the two datasets confirmed our hypothesis that the size of the learning dataset influences the training model.

Discussion:

In the context of our thesis titled "Comparing Data Presentation Techniques for CNN-based IDS," we derived interesting observations from the four tests conducted. For instance, in test N1, we found that using different image and frame sizes with both datasets revealed the significant impact of specific configuration parameters. Subsequently, we continued testing and comparing results with both datasets until we concluded that the best combination for our tests was (11×11/64/true). This indicates that a frame size of 11x11, an image size of 64, and enabling resampling (true) yielded the optimal results. Furthermore, the learning curves for both NSL-KDD and CICIDS2017 datasets demonstrated that our CNN model achieved ideal training accuracy, with the loss falling within a relatively low range, thus confirming its high training capacity.

Test N3 aimed to investigate whether the learning dataset size influenced the results. We observed that the dataset type indeed impacted the quality, as expected. Notably, in the case of NSL-KDD, the precision decreased from 0.9920 to 0.9880 and remained at that level. Conversely, for CICIDS2017, the precision dropped from the highest value of 0.9940 to the lowest at 0.9810, providing clear evidence of the quality differences resulting from distinct datasets.

Chapter 4: Implementation, Result and Discussion

In the final test, we sought to verify the effectiveness of our CNN model using a different classifier, namely Random Forest. The obtained results confirmed our hypothesis, as the precision dropped from 0.99% with our CNN model to 0.88% with Random Forest. Additionally, this further supported our previous assertion that the size of the learning dataset influences the training model.

4.8 Conclusion

This chapter was dedicated to the implementation, results, and discussion. We discussed the CNN architecture, the tools used to achieve it, and the testing protocol, which outlined the essential steps followed to test our model. These steps included data preparation, pseudo-image generation, CNN configuration, training phase, and testing. We conducted a total of four tests to confirm or refute our hypotheses. Each test provided us with valuable insights. For example, test N2 enabled us to determine the best choice for our model through a comparison. Test N3 confirmed our hypothesis that the type of dataset impacts the results of the learning size. In the final test, we compared the CNN and Random Forest classifiers to determine which one was more powerful, reliable, and efficient in our case. We used a Confusion Matrix analysis to make this determination.

General Conclusion

General Conclusion

In conclusion, this thesis has examined the effectiveness of different data presentation techniques in the context of CNN-based Intrusion Detection Systems (IDS).

What this work represents is an answer to the problematic passing through different steps:

First we defined the technology of IDS, for the protection of computer systems, and how Deep Learning is an effective and promising technique for the development of IDS.

Secondly, presented the concept of CNN based IDS and find out how to apply it to our research topic. We have developed a protocol for tests to study the different transformations of non-image data into pseudo-image, and their impact on the quality of IDS-CNN.

The results proved to us that transforming IDS data into a pseudo-image gives better results for a CNN, whereas many previous studies using CNN as IDS presented the data as a simple vector.

We found the right transformations, to harness the power of CNN, and thus produce an efficient IDS. Transforming the data to a Square image and with Resamling is the best transformation, which produces optimal performance.

The CNN kept its strength and provided us with some powerful, optimal results even though we came to a conclusion that the quality differentiate from dataset to another. To make this allegation more strong and proved , another classifier were employed , Random_Forest which itself is a strong one as well , the results displayed only affirmed our hypothesis that how DL(Deep Learning) approaches are a defensive weapon against malicious activities and how CNN is that scalable and reliable in our case.

Finally this modest work is a result of research done by us and mentored by our dear supervisor, and it's always open for any new improvements that we can list in this area as perspectives for the future

Future Perspective

Our discoveries during this work have opened new research doors in detail below:

- To improve the capability of CNN-based IDS, we intend to study hybrid CNN and RNN system, since this type of Hybrid IDS can be used to handle spatial and temporal features simultaneously, resulting in high detection capability, and a low rate of false positive alarms.

General Conclusion

- Do tests on other datasets, and on other types of networks such as Software-Defined Network (SDN).

Bibliography

[1] Wood, Mark and Erlinger, Michael. Intrusion detection message exchange requirements. https://tools.ietf.org/html/rfc4766, 2002. [online ; Consulted in Fabruary,15th,2020].

[2] Anomaly detection in wireless sensor networks: a survey Journal of Network and Computer Applications (2011) Wood, Mark and Erlinger, Michael. Intrusion detection message exchange requirements.
[3] Hervé Debar, Marc Dacier, and Andreas Wespi. A revised taxonomy for intrusion detection systems. In Annales des télécommunications, volume 55, pages 361–378.Springer, 2000.

[4] Nasrin Sultana, Naveen Chilamkurti, Wei Peng, and Rabei Alhadad. Survey on sdn based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications, 12(2) :493–501, 2019.

[5] Liran Lerman, Olivier Markovitch, and Gianluca Bontempi. system detection intrusion based on machine learning. PhD thesis, thesis of doctorat, University libre de Bruxelles, 2008.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE,2017.

[7] Soman KP, Mamoun Alazab, et al. A comprehensive tutorial and survey of applications of deep learning for cyber security. 2020.

[8] Sumeet Dua and Xian Du. Data mining and machine learning in cybersecurity. Auerbach Publications, 2016.

[9] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, and Alfredo De Santis. Network anomaly detection with the restricted boltzmann machine. Neurocomputing, 122 :13–23, 2013.

[10] Mehedy Masud, Latifur Khan, and Bhavani Thuraisingham. Data mining tools for malware detection. Auerbach Publications, 2016.

[11] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. Neurocomputing, 234 :11–26, 2017.

[12] Hassan Hadi Al-Maksousy, Michele C Weigle, and Cong Wang. Nids : Neural network based intrusion detection system. In 2018 IEEE International Symposium on Technologies for Homeland Security (HST), pages 1–6. IEEE, 2018.

[13] Clément Dalloux, Natalia Grabar, and Vincent Claveau. Détection de la négation :corpus français et apprentissage supervisé. Revue des Sciences et Technologies de l'Information-Série TSI : Technique et Science Informatique, pages 2019.

[14] Frank Rosenblatt. The perceptron : a probabilistic model for information storage andorganization in the brain. Psychological review, 65(6) :386, 1958.

[15] Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville published by MIT Press, 2016.

Bibliography

[16] Mohammadpour, L. et al. (2022) A survey of CNN-based network intrusion detection, MDPI. Available at: https://www.mdpi.com/2076-3417/12/16/8162 .
[17] Classification accuracy Classification Accuracy - an overview | ScienceDirect Topics. Available at: https://www.sciencedirect.com/topics/engineering/classification-accuracy .

[18] Mohammadpour, L. et al. (2020) A mean convolutional layer for intrusion detection system, Security and Communication Networks. Available at: https://www.hindawi.com/journals/scn/2020/8891185/ (Accessed: 26 May 2023).

[19] Sample input using the different preprocessing methods for the same ... Available at: https://www.researchgate.net/figure/Sample-input-using-the-different-preprocessing-methods-for-the-same-data-point-a_fig1_352502419 .

[20] Brownlee, J. (2019) Python ecosystem for Machine Learning, MachineLearningMastery.com. Available at: https://machinelearningmastery.com/python-ecosystem-machine-learning .

[21] What is python? executive summary Python.org. Available at: https://www.python.org/doc/essays/blurb/ .

[22] E L O D E E S WE BELIEVE IN CONSCIOUS AND 'STRONG' ARTIFICIAL INTELLIGENCE. Available at: https://elodees.com/EN.Several-frameworks-to-move-to-reinforcement-learning .

[23] Team, K. Keras documentation: Why choose keras?, Keras. Available at: https://keras.io/why_keras/.

[24] Features in CIC ids 2017 dataset | download scientific diagram. Available at: https://www.researchgate.net/figure/Features-in-cic-ids-2017-dataset_tbl1_343850781

ملخص

كان الهدف من هذه الدراسة هو اختبار فعالية تقنيات تصور البيانات المختلفة في سياق أنظمة كشف التسلل القائمة على الشبكة العصبية التلافيفية، أظهر التطور السريع للتكنولوجيا والمخاطر المرتبطة بها أن التدابير الأمنية التقليدية لحماية الملفات والبيانات والمعلومات الشخصية غير كافية. لحل هذه المشكلة ، تم تنفيذ حلول نظام كشف التسلل / نظام منع التسلل لتأمين الشبكة والحماية من نقاط الضعف. ومع ذلك ، فإن ظهور أساليب التعلم العميق ، بما في ذلك الشبكات العصبية التلافيفية، قد أوجد طرقا جديدة لمواجهة تحديات الأمن السيبراني. ركزت هذه الدراسة على تقنيات تمثيل البيانات غير المصورة عن طريق تحويلها إلى تمثيلات مرئية لاستغلال قدرات التعرف على الصور الكامنة في الشبكات العصبية التلافيفية. من خلال فحص تقنيات عرض البيانات المختلفة ، يهدف البحث إلى تحسين أداء نظام كشف التسلل. تم استخدام مجموعتي بيانات مختلفتين للتجارب: مجموعة بيانات NSL-KDD ، والتي غالبا ما تستخدم كمعيار في أبحاث كشف التسلل ، ومجموعة بيانات CICIDS2017. من خلال تحليل مقارن ، قارنت الدراسة نتائج الصور المختلفة وتقنيات التصنيف المستندة إلى الشبكة العصبية التلافيفية مع Random Forest ، بهدف تحديد تقنية عرض البيانات الأكثر لنظام كشف التسلل القائم على الشبكة العصبية التلافيفية. توفر نتائج هذه الدراسات معلومات قيمة لتحسين أساليب نظام كشف التسلل القائم على الشبكة العصبية التلافيفية ، مما يضمن الدقة ووقت التنفيذ القصير. بشكل عام ، تساهم هذه الدراسة في مجال أمن المعلومات من خلال تسليط الضوء على دور تقنيات عرض البيانات في أنظمة نظام كشف التسلل القائم على الشبكة العصبية التلافيفية. من خلال فهم نقاط القوة والضعف في الأساليب المختلفة ، يمكن للمؤسسات تحسين تدابير أمان الشبكة واتخاذ قرارات مستنيرة بشأن تنفيذ حلول نظام كشف التسلل.

الكلمات المفتاحية: الشبكة العصبية التلافيفية، عرض البيانات، نظام كشف التسلل / نظام منع التسلل، NSLKDD. CICIDS2017 ، Random Forest.

Abstract

This study aimed to investigate the effectiveness of different data presentation techniques in the context of CNN-based Intrusion Detection Systems (IDS). The rapid evolution of technology and associated risks have highlighted the inadequacy of traditional security measures in safeguarding files, data, and personal information. To address this, IDS/IPS solutions have been employed to secure networks and protect against vulnerabilities. However, the emergence of deep learning methods, including Convolutional Neural Networks (CNNs), has provided new possibilities for addressing cybersecurity challenges.This study focused on techniques for presenting non-image data by transforming it into visual representations for exploiting the inherent image recognition capabilities of CNNs. By exploring various data presentation techniques, the research aimed to improve the performance of IDS. Two distinct datasets were utilized for experimentation: the NSL-KDD dataset, widely employed as a benchmark in intrusion detection research, and the CICIDS2017 dataset.Through comparative analysis, the study compared the results obtained from different image frames and CNN-based classification techniques with Random Forest, aiming to determine the most effective data presentation technique for CNN-based IDS. The findings of this research provide valuable insights into optimizing CNN-based IDS approaches, ensuring precision and short execution time.Overall, this study contributes to the field of cybersecurity by shedding light on the role of data presentation techniques in CNN-based IDS. By understanding the strengths and weaknesses of various approaches, organizations can enhance their network security measures and make informed decisions regarding the deployment of IDS solutions

.**Keywords:** CNN, Data presentation, IDS/IPS, NSLKDD,CICIDS2017, Random Forest .

Résumé

Ce mémoire visait à étudier l'efficacité de différentes techniques de présentation des données dans le contexte des systèmes de détection d'intrusions (IDS) basés sur les réseaux de neurones convolutionnels (CNN). L'évolution rapide de la technologie et les risques associés ont souligné l'insuffisance des mesures de sécurité traditionnelles pour protéger les fichiers, les données et les informations personnelles. Pour remédier à cela, des solutions IDS/IPS ont été mises en place pour sécuriser les réseaux et se prémunir contre les vulnérabilités. Cependant, l'émergence de méthodes d'apprentissage en profondeur, notamment les CNN, offre de nouvelles possibilités pour relever les défis de la cybersécurité. Cette étude s'est concentrée sur les techniques de présentation des données non graphiques en les transformant en représentations visuelles pour exploiter les capacités inhérentes de reconnaissance d'images des CNN. En explorant différentes techniques de présentation des données, la recherche visait à améliorer les performances des IDS. Deux ensembles de données distincts ont été utilisés pour les expérimentations : le jeu de données NSL-KDD, largement utilisé comme référence dans la recherche sur la détection d'intrusions, et le jeu de données CICIDS2017. Par le biais d'une analyse comparative, l'étude a comparé les résultats obtenus à partir de différentes structures d'images et de techniques de classification basées sur les CNN avec Random Forest, dans le but de déterminer la technique de présentation des données la plus efficace pour les IDS basés sur les CNN. Les conclusions de cette recherche fournissent des connaissances précieuses pour optimiser les approches IDS basées sur les CNN, en garantissant une précision et un temps d'exécution courts. Dans l'ensemble, cette étude contribue au domaine de la cybersécurité en mettant en lumière le rôle des techniques de présentation des données dans les IDS basés sur les CNN. En comprenant les forces et les faiblesses des différentes approches, les organisations peuvent améliorer leurs mesures de sécurité réseau et prendre des décisions éclairées quant au déploiement de solutions IDS.

Mots-clés: CNN, Data presentation, IDS/IPS, NSLKDD, CICIDS2017, Random Forest .