جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES



كلية العلوم والتقنيات فاس
FACULTÉ DES SCIENCES ET TECHNIQUES DE FÈS

Centre d'Etudes Doctorales: Sciences et Techniques de l'Ingénieur

DOCTORAL THESIS DEFENSE PRESENTED BY

Ms. Sabrine Ennaji

Discipline: Computer Science
Specialty: Computer Science

Thesis Subject

Cognitive Security Using Machine Learning Techniques

Doctoral Program in Engineering, Physical Sciences, Mathematics and Computer Science

Thesis presented and defended on Saturday, December 23, 2023 at 10 a.m. at the National School of the Applied Sciences of Fez, in front of the members of the examination panel composed of:

Full Name	Title	Institution	Examination
ZARGHILI Arsalane	PES	Faculté des Sciences et Techniques-Fez	President
BENSLIMANE Mohamed	PH	École Supérieure de Technologies -Fez	Reviewer
EZZOUHAIRI Abdellatif	PES	École Nationale des Sciences Appliquées-Fez	Reviewer
LAZAAR Mohamed	PH	École Nationale Supérieure d'Informatique et d'Analyse des Systèmes-Rabat	Reviewer
ZOUHAIR Abdelhamid	PH	Faculté des Sciences et Techniques de Tanger	Examiner
KENZI Adil	PES	École Nationale des Sciences Appliquées de Fès	Examiner
BOULAALAM Abdelhak	PH	École Nationale des Sciences Appliquées de Fès	Examiner
HADDOUCH Khalid	PH	École Nationale des Sciences Appliquées de Fès	Supervisor
EL AKKAD Nabil	PH	École Nationale des Sciences Appliquées de Fès	Co-Supervisor

Research Laboratory: Engineering, Systems and Applications

Institution: École Nationale des Sciences Appliquées de Fès

Résumé

Avec la dépendance croissante aux réseaux informatiques, les systèmes de détection d'intrusion sur réseau (NIDS) sont devenus essentiels pour garantir la sécurité des systèmes en réseau. Les approches traditionnelles basées sur des règles sont souvent limitées dans la détection des attaques sophistiquées et évolutives. Pour relever ce défi, les techniques d'apprentissage automatique ont retenu une attention importante pour améliorer les performances des systèmes de détection d'intrusion. Cette thèse présente un aperçu de diverses techniques intelligentes utilisées dans les systèmes de détection d'intrusion sur les réseaux. Il commence par discuter des concepts fondamentaux de la détection des intrusions sur les réseaux et des défis associés à la détection des attaques modernes. Par la suite, il explore l'application d'algorithmes d'apprentissage automatique, tels que Random Forest (RF), Naïve Bayes (NB), Support Vector Machines (SVM), les réseaux de neurones et les méthodes d'ensemble, pour construire (un) (système) efficace pour) NIDS. De plus, il met en évidence les avantages et les inconvénients de chaque algorithme. Il aborde également les méthodes de sélection et d'extraction de fonctionnalités qui jouent un rôle essentiel dans l'amélioration des performances des NIDS basés sur l'apprentissage automatique. De plus, une nouvelle approche basée sur la technique de validation croisée imbriquée est proposée afin de résoudre le problème de la gestion des ensembles de données déséquilibrés, qui est un problème courant dans la détection d'intrusion et présente des techniques permettant d'atténuer son impact sur la précision de la détection. Ce travail examine l'importance de la qualité des données de formation et discute des approches permettant de générer des ensembles de données réalistes et diversifiés afin de former efficacement des modèles d'apprentissage automatique. Il explore également le rôle de l'ingénierie des fonctionnalités et l'utilisation des connaissances du domaine pour améliorer les capacités de détection du NIDS. Enfin, cette thèse discute et explore de nouveaux défis et des directions de recherche ouvertes dans le domaine de l'apprentissage profond pour les systèmes de détection d'intrusion sur les réseaux. Il souligne la nécessité de développer des NIDS adaptatifs et résilients, capables de détecter efficacement les nouvelles attaques.

Mots clés: systèmes de détection d'intrusion réseau, machine learning, deep learning, validation croisée imbriquée, ingénierie de fonctionnalités.

Abstract

With the increasing reliance on computer networks, network intrusion detection systems have become crucial for ensuring the security of networked systems. Traditional rule-based approaches are often limited in detecting sophisticated and evolving attacks. To address this challenge, machine learning techniques have gained important attention for enhancing the performance of intrusion detection systems. This thesis presents an overview of various intelligent techniques used in network intrusion detection systems. It begins by discussing the fundamental concepts of network intrusion detection and the challenges associated with detecting modern-day attacks. Subsequently, it explores the application of machine learning algorithms, such as Random Forest (RF), Naïve Bayes (NB), Support Vector Machines (SVM), neural networks, and ensemble methods, for building an effective NIDS. Furthermore, it highlights the pros and cons of each algorithm. It also discusses feature selection and extraction methods that play a critical role in improving the performance of machine learning-based NIDS. Moreover, a novel approach based on the Nested Cross Validation technique is proposed in order to address the issue of handling imbalanced datasets, which is a common problem in intrusion detection, and presents techniques for mitigating its impact on detection accuracy. This work examines the importance of training data quality and discusses approaches for generating realistic and diverse datasets to train machine learning models effectively. It also explores the role of feature engineering and the use of domain knowledge to enhance the detection capabilities of NIDS. Finally, this thesis discusses and explores new challenges and open research directions in the field of deep learning for network intrusion detection systems. It emphasizes the need for developing adaptive and resilient NIDS that can effectively detect novel attacks.

Keywords: network intrusion detection systems, machine learning, deep learning, nested cross-validation, feature engineering.

Acknowledgemengts

This Ph.D. has been an enriching experience for me, despite all the ups and downs I encountered, but it was notably full of wealthy lessons. My first gratitude goes to my supervisors, Professor Khalid Haddouch and Professor Nabil El Akkad, for allowing me to join their research team and work on a topic of great relevance with scientific rigor. I thank them for their trust in me and their assistance over these four years.

I would also like to thank the jury members of my thesis defense. I express all my gratitude to Professor Mohamed Lazaar, Professor Abdellatif Ezzouhairi, Professor Mohamed Benslimane, Professor Abdelhamid Zouhair, Professor Adil Kenzi and Professor Abdelhak Boulaalam who agreed to read and review my thesis. Special thanks to the President Professor Arsalane Zarghili. I thank all of them for their time and attention to my work, as well as their insightful feedback. It truly is an honor to have all of these experts examine my work.

I would like to express my gratitude to the research members of the Cybersecurity Laboratory at The University of Sapienza in Rome, Italy who welcomed me, and special thanks to Professor Luigi Mancini for his trust as he gave me the opportunity to improve my knowledge, develop new skills and demonstrate my abilities in the domain of cybersecurity. This experience was extremely challenging and I find it as additional value for my research career. Moreover, I had a chance to have such a friendly atmosphere created by this team as it contributed to having a good mood and working comfortably in the office.

Dedication

To my loving **Parents** and supportive sisters Hind and Jihane,

This thesis is dedicated to you, the pillars of my life, whose unwavering love, encouragement, and guidance have propelled me forward on this academic journey. Your belief in my abilities, endless sacrifices, and steadfast support has been the foundation upon which I built this work.

Mom and **Dad**, your tireless efforts to provide me with the best education and opportunities have shaped the person I am today. Your constant encouragement, wisdom, and belief in my potential have fueled my determination to excel in my studies. Your sacrifices and unwavering love have been my driving force, pushing me to strive for excellence.

To my sisters, who have always been my biggest cheerleaders and confidantes, thank you for your unwavering support and understanding. Your belief in my abilities, your words of encouragement, and your presence during both the challenging and triumphant moments have given me the strength to overcome obstacles and persevere.

This thesis is a testament to your love, sacrifice, and unwavering support. Without you, this accomplishment would not have been possible. As I celebrate this milestone, I want to express my deepest gratitude for your belief in me and for being the driving force behind my academic pursuits.

With all my love and appreciation,

Table of Contents

Ms. Sabrine Ennaji..... 1

Résumé.....2

Abstract.....3

List of Acronyms.....9

List of Figures.....9

List of Tables.....10

General Introduction.....11

Chapter I.....15

Machine Learning: Background and its Applications.....16

1.1 Introduction..... 16

1.2 Machine Learning Approaches..... 17

1.2.1 Definition and the Important Applications of Machine Learning..... 18

1.2.2 Machine Learning Techniques..... 19

1.2.2.1 Supervised Learning..... 20

1.2.2.2 Semi-Supervised Learning..... 31

1.2.2.3 Unsupervised Learning..... 32

1.2.2.4 Reinforcement Learning..... 34

1.2.3 Deep Learning..... 35

1.3 Machine Learning Applications..... 37

1.4 Advantages and Drawbacks of Machine Learning..... 38

1.5 Conclusion..... 39

Chapter II.....40

Machine Learning for Cybersecurity.....40

2.1 Introduction..... 40

2.2 The Importance of Intrusion Detection..... 43

2.2.1 Security Protection Techniques..... 43

2.2.2 Why Intrusion Detection Systems?..... 46

2.3 Intrusion Detection Systems Fundamentals....47
2.3.1 Definition and Purpose of Intrusion Detection Systems....47
2.3.2 Components of an Intrusion Detection System....49
2.3.3 Ways to Detect Intrusions....51
2.4 Types of Intrusion Detection Systems....53
2.4.1 Network-Based Intrusion Detection Systems (NIDS)....53
2.4.2 Host-Based Intrusion Detection Systems (HIDS)....54
2.4.3 Cloud-Based Intrusion Detection Systems (Cloud-based IDS)....55
2.5 Limitations of Intrusion Detection Systems....57
2.6 Intelligent Network Intrusion Detection Systems....58
2.7 Challenges and Considerations in Intelligent NIDSs....60
2.8 Conclusion....62
Chapter III....62
A Powerful Ensemble Learning Approach for Improving Network Intrusion Detection System (NIDS)....62
3.1 Introduction....63
3.2 Related Works....65
3.3 Comparative Study....69
3.4 Proposed Model....73
3.4.1 Dataset Description....73
3.4.2 Experimental Setup....75
3.4.3 Methodology....76
3.4.4 Evaluation of the Model....78
3.5 Results and Discussion....80
3.6 Conclusion and Future Scope....82
Chapter IV....84
Importance of Hybridization Modeling for Strengthening the Network Intrusion Detection 84
4.1 Introduction....84
4.2 Related works....85
4.2.1 Single learning method....85
4.2.2 Collaborative learning method....87
4.3 Proposed Hybridization Modeling....89

4.4 Results and Discussion....91
4.5 Conclusion....93
Chapter V....94
i-2NIDS Novel Intelligent Intrusion Detection Approach for a Strong Network Security, Dealing with Imbalanced Classification Issue....94
5.1 Introduction....94
5.2 Related Works....96
5.3 The Proposed i-2NIDS and Methodology....99
Data Preprocessing....100
Feature Selection....101
Data pre-processing for training dataset based on nested cross-validation....102
Performance Evaluation....103
5.4 Experiments and Discussion of Results....104
Tools and Environment....104
Experimentation Results....105
5.5 Comparison with existing methods....109
5.6 Conclusion....110
General Conclusion and New Challenges....112
Scientific Publications....114
A Powerful Ensemble Learning Approach for Improving Network Intrusion Detection System....114
Importance of Hybridization Modeling for Strengthening the Network Intrusion Detection 114
i-2NIDS Novel Intelligent Intrusion Detection Approach for a Strong Network Security, Dealing with Imbalanced Classification Issue....114
References....115

List of Acronyms

AI	Artificial Intelligence
ML	Machine Learning
IT	Internet Technology
DL	Deep Learning
RF	Random Forest
LG	Logistic Regression
KNN	K-Nearest Neighbor
MLP	Multilayer Perceptron
NB	Naïve Bayes
DT	Decision Trees
ANN	Artificial Neural Networks
IoT	Internet of Things
IDS	Intrusion Detection System
NIDS	Network Intrusion Detection System
NLP	Natural Language Processing
MSE	Mean Squared Error
SVM	Support Vector Machines
CV	Computer Vision
PCA	Principal Component Analysis
HIDS	Host-based Intrusion Detection Systems
AE	Auto-Encoder
AdaBoost	Adaptive Boosting

List of Figures

Figure 1. Linear Regression Vs Logistic Regression (Credits to: DataCamp).....22

Figure 2. MultiLayer Perceptron Layers (Credits to: Analytics Vidhya).....29

Figure 3. Distribution of detected cyberattacks worldwide in 2022, by type (Credits to: Statista.com).....41

Figure 4. Intrusion Detection System (IDS) Architecture.....48

Figure 5. Components of an IDS (Credits to: TechGenix.com).....51

Figure 6. Network Intrusion Detection System (NIDS) (Credits to: Comparitech.com).....54

Figure 7. Host-Based Intrusion Detection Systems (HIDS) (Credits to: Comparitech.com).....55

Figure 8. Example of a Cloud-Based IDS (Credits to: Blumira.com).....56

Figure 9. Machine Learning-based IDS.....58

Figure 10. Comparison of Machine Learning algorithms using the most known indicators.....72

Figure 11. Time of learning (Seconds) and prediction (Milliseconds).....73

Figure 12. Google Colab Logo (Credits to: Google colab).....76

Figure 13. A conceptual framework for an Intrusion Detection System models.....77

Figure 14. Truth table of the confusion matrix.....78

Figure 15. Comparison of the five proposed ensemble learning models using the most known indicators (%).....81

Figure 16. Accuracy rate of the proposed method and other existing approaches.....82

Figure 17. Prediction time of the 5 proposed ensemble learning models in Seconds.....82

Figure 18. Flow chart representing the proposed methodology.....90

Figure 19. Time required by each algorithm for the prediction process in Ms.....92

Figure 20. Comparison of the proposed hybrid model to another existing model (%).....92

Figure 21. The architecture of the proposed model.....100

Figure 22. The performance metrics of the proposed models for normal class identification (%).....108

Figure 23. The performance metrics of the proposed models for the 4 attack classes (DoS, Probe, U2R and R2L) identification (%).....108

Figure 24. The accuracy rate for testing results for binary classification.....109

Figure 25. The performance compared with other methods using the NSL-KDD dataset.....110

List of Tables

Table 1. Presentation of the AI strength and weak points.....39

Table 2. Detection ways of an IDS.....53

Table 3. Comparison of Machine Learning algorithms using NSL-KDD dataset in % (Test results).....71

Table 4. Comparison in terms of Learning and prediction time of Machine Learning algorithms using NSL-KDD data set.....72

Table 5. Comparison with the five proposed ensemble learning models using NSL-KDD data set in % (TEST RESULTS).....81

Table 6. The performance of the proposed hybrid IDS model compared to single classifiers over the test set (%).....91

Table 7. Types of cross-validation technique.....95

Table 8. Attack class distribution in the NSL-KDD dataset.....101

Table 9. Top 10 most important features selected in the NSL-KDD dataset.....102

Table 10. Confusion matrix for 5-class classification.....103

Table 11. The performance metrics of the proposed models for a multi-class classification detecting Normal class (Test results %).....106

Table 12. The performance metrics of the proposed models for a multiclass classification detecting DoS/DDoS Attack (Test results %).....107

Table 13. The performance metrics of the proposed models for a multiclass classification detecting probe Attack (Test results %).....107

Table 14. The performance metrics of the proposed models for a multiclass classification detecting r2l Attack (Test results %).....107

Table 15. The performance metrics of the proposed models for a multiclass classification detecting U2R Attack (Test results %).....107

General Introduction

The popularity of public networks, especially the Internet, and the unprecedented growth in their users has raised major security concerns, especially in terms of users' privacy **[1]**. The risks of violating users' privacy exacerbate when we consider the growing trend towards using smart devices, the emergence of cloud services, and the interest in mobility solutions **[2]**. These trends require full collaboration between clients, service providers, and web hosts to form a solid foundation for a secure environment. This environment should implement sufficient security measures that can prevent intruders from performing a diverse range of malicious actions **[3]**. This evolution has resulted in the development of new strategies for gaining access to networks and information systems in order to facilitate transactions. Intrusion detection is a critical step in network security. Despite the flexibility of being placed in critical areas, the real-time detection, and quick reaction times that an Intrusion Detection System (IDS) can provide for the analysis of both inbound and outbound traffic, its performance still needs to be improved. This is normally due to the fact that network traffic is increasing dramatically every day as a result of the widespread use of cloud services and the Internet of Things (IoT). It makes it difficult for an IDS to discern between regular and abnormal network traffic behavior, especially when it comes to the detection of zero-day attacks **[4]**.

Artificial intelligence approaches are increasingly being employed nowadays to reinforce and improve assault detection rates. Their significant potential is considered as one of the best solutions to deal with the limitations of Intrusion Detection Systems (IDSs), as Machine Learning (ML) and Deep Learning (DL) mechanisms have a great ability to identify trends and patterns without any human intervention **[5]**. Unlike the IDSs that frequently need someone to administrate them. However, with the endless current security challenges, it will never be possible to have "the perfect" model. There are many loopholes that need to be addressed, such as overfitting and underfitting problems, bias and discrimination, data quality, highly costly computations, etc. Moreover, different novel cyberattacks can be a serious threat to intelligent systems, which aims to trick the latter. This is still an open research problem that is slightly taken into consideration.

The objective of this research thesis is to develop new techniques to cope with the shortcomings of the existing intelligent Intrusion Detection Systems and build robust models that take into account almost all the present-day security risks that have been mentioned previously. These objectives can be summarized in two main categories:

- **Strengthen:** Enhancing and updating the performance of an Intelligent Intrusion Detection System (IDS) based on the proposition of a novel approach, which tackles the weaknesses of such models.
- **Exploring new challenges:** Venturing into novel problematics in this context and making them more under limelight in order to urge other researchers to contribute to this topic and put forward some strategies to overcome them.

The main research contributions of this thesis are as follows:

1. Comparative analysis of several machine learning algorithms, which was a methodical process for selecting the best algorithm for enhancing the performance of network intrusion detection systems, and learning more about how various algorithms react in various situations. It is an essential step in the creation and use of machine learning models.
2. Based in this comparative analysis study, we have developed a powerful Network Intrusion Detection System (NIDS) based on the ensemble learning strategy, whereby it makes five machine learning algorithms in collaboration, aiming to classify the network traffic into normal and abnormal behaviors. Our method reached the best results, compared to existing models opting for almost the same techniques and dataset.
3. We presented also an analytical study that shows the importance of AdaBoost algorithm hybridization approach in maximizing the accuracy rate of a NIDS in distinguishing between a malicious activity and a normal one. The discussion of this study provides a full understanding of the upsides and downsides of each classifier used, in addition to the different types of learning *(i.e., single learning Vs ensemble learning)*. Consequently, it is a good reference for other researchers to contribute more to this field.
4. We introduced a novel methodology based on the nested cross-validation technique, which has a significant ability to successfully tune the parameters of a machine learning algorithm and concurrently generalize the error estimate. Furthermore, it deals with the imbalance issue. The obtained results validate the well-performance and efficiency that our model achieved.

The remainder of this thesis is organized as follows: In Chapter **1**, we present a full background of Machine Learning (ML) and its application in several domains, including the cybersecurity field that belongs to our research interest. Almost all the types of Machine Learning have been introduced, following with some examples that we relied on in our research

project. Moreover, we also mention the upsides and downsides of Machine Learning. Then, in Chapter **2**, we present in detail how Machine Learning tools have been able to cope with the limitations of Intrusion Detection Systems (IDSs) and how they have been integrated in the latter, with showing the importance of intrusion detection task in general. Furthermore, the environments that require a strong security have been depicted and defined. Chapter **3** introduces our proposed ensemble learning method using different Machine Learning algorithms to improve the intrusion detection task. Chapter **4** describes an extended version of the first contribution about the importance of multi-model learning. In Chapter **5,** we propose a novel intelligent system that we called i-2NIDS, which is an enhanced version of a traditional Network Intrusion Detection System (NIDS). Finally, we conclude our thesis report and provide new challenges to Machine Learning and Deep Learning based systems. In addition to future research directions.

Chapter I

Machine Learning: Background and its Applications

1.1 Introduction

The goal of artificial intelligence *(AI)*, a branch of computer science and engineering, is to build intelligent computers that are capable of carrying out activities that need human-like intelligence **[6]**. The ultimate purpose of AI lies in creating machines that, without direct human input, comprehend natural language, and adapt to new circumstances in order to solve difficult and critical problems **[7]**.

The foundation of artificial intelligence *(AI)* is the idea of imitating human intelligence in computers, allowing them to sense their surroundings, think about them, and take action to accomplish predetermined goals **[8]**. AI is an interdisciplinary field that incorporates ideas and methods from many other disciplines, including computer science, mathematics, neurology, psychology, and engineering **[9]**. Within AI, there are various crucial elements and methods as described below:

(i) **Machine Learning:** A subset of artificial intelligence *(AI)*, it enables machines to learn from data and enhance their performance over time. Large datasets are used to

train algorithms, enabling computers to spot patterns, forecast the future, and take decisions without explicit programming **[10]**.

(ii) **Deep Learning:** The structure and operation of the human brain served as the inspiration for this group of algorithms. They help deep learning, a potent Machine Learning method, perform challenging jobs like speech and picture recognition **[11]**.

(iii) **Computer Vision:** Giving machines the ability to decipher and comprehend visual data from pictures and videos is the focus of computer vision. It is necessary for operations like face recognition, object detection, and autonomous vehicles **[12]**.

(iv) **Robotics:** AI is essential to robotics because it allows machines to interact with the real world, understand their surroundings, and carry out activities on their own **[13]**.

(v) **Natural Language Processing:** Machines can now comprehend, interpret, and produce human language thanks to a process called natural language processing (NLP). Applications like voice assistants, chatbots, and language translation are made possible by this **[14]**.

Numerous industries, including healthcare, computer security, banking, education, transportation, gaming, and more have benefited from the use of AI. With continued research and development, it is still developing quickly and pushing the limits of what AI is capable of **[15]**. Although AI offers many advantages and potential, it also poses significant ethical, privacy, and societal issues. Researchers, governments, and industry stakeholders are heavily focused on ensuring that AI technologies are created and implemented ethically **[16]**.

In this research, we are opting for different Machine Learning approaches to deal with cybersecurity threats.

1.2 Machine Learning Approaches

This section presents the definition of ML techniques and its most important applications, including network security and its different techniques with some examples that we employed in our research project.

1.2.1 Definition and the Important Applications of Machine Learning

The field of machine learning in artificial intelligence (AI) focuses on developing algorithms and models that enable computers to learn and form predictions or judgments without needing to be explicitly programmed. It is based on the idea that systems can independently gain knowledge from their experiences and develop. Applications for machine learning have grown in importance as a result of their capacity to analyze massive volumes of data, extract insightful information, and automate difficult processes **[17]**.

The importance of machine learning applications lies in different major factors:

- **Efficiency and Automation:** Machine learning enables the automation of time-consuming, repetitive processes. Organizations can free up human resources to work on more difficult and innovative projects by training models to carry out these duties. Machine learning-based automation also lowers the possibility of human error, increasing productivity and efficiency.
- **Understanding and Processing of Natural Language:** Machine learning is important for understanding and processing natural language (NLP). It makes it possible for chatbots, virtual helpers, and voice recognition technology to fully understand and react to human language. Machine learning models can translate languages, extract useful

information from unstructured data, and analyze text sentiment, resulting in more effective and seamless communication.

- **Data-driven Decision Making:** Machine learning algorithms are capable of accurately and swiftly analyzing and interpreting enormous datasets. Machine learning enables businesses and organizations to make data-driven choices by revealing patterns, trends, and relationships in data. Better strategic planning, resource allocation, and efficiency can result from this.
- **Medicine and healthcare:** Machine learning has the power to completely transform these industries. It can help with disease diagnosis, patient outcome prediction, and therapy choice identification. Medical imaging data from X-rays and MRIs can be analyzed using machine learning algorithms to help find anomalies. In order to help medical personnel make better decisions and enhance patient care, they can also mine electronic health records for trends and risk factors.
- **Cybersecurity and fraud detection:** By examining massive volumes of data and spotting anomalies, Machine Learning algorithms are able to recognize patterns of fraudulent conduct. Having this ability is essential for identifying network invasions, credit card fraud, and other security breaches. Machine learning models are useful tools for protecting private data and averting financial losses because they can learn and adapt to new fraud tendencies.

These can be considered as only a few illustrations of how crucial machine learning applications are. This new technology is continuing to alter industries, enhance processes, and open up new opportunities across a variety of fields as a result of technological breakthroughs and the accessibility of enormous amounts of data.

1.2.2 Machine Learning Techniques

Machine learning algorithms come in a variety of forms, each with a unique methodology and traits. In the following subsections, we present the main categories:

1.2.2.1 Supervised Learning

A machine learning technique called supervised learning uses labeled training data to teach a system how to make predictions or categorize data. In supervised learning, target variables or related output labels are linked to the input data, also known as features **[18]**.

Building a model that can generalize and make precise predictions on unknown or newly created input data is the aim of supervised learning **[19]**. This model is built on the patterns and relationships that it acquired from the labeled training samples. Following are some popular examples of the supervised learning algorithms that are widely used:

A. Linear Regression

It is employed to forecast continuous numerical values. The goal variable and the input features are established to be linearly related. The algorithm makes the assumption that the independent variables (features) and the dependent variable (goal) are correlated linearly **[20]**. The main ideas and characteristics of linear regression are as follows:

- **Simple linear regression:** When it is about only one input feature (independent variable) and one target variable (dependent variable) this type of regression is used **[21]**. It seeks to identify the line that most accurately depicts the connection between the feature and the target.

$$Y = mx + b, \quad (1)$$

where y is the anticipated target variable, x is the input feature, m is the line's slope, and b is the y-intercept, which defines the line.

- **Multiple Linear Regression:** When there are numerous input features, multiple linear regression is performed. In order to accommodate numerous independent variables, it expands ordinary linear regression. The algorithm aims to determine the weights or coefficients that each feature should have in order to accurately forecast the target variable **[22]**. As a result, the equation is written as:

$$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n, \qquad (2)$$

where b_0 is the y-intercept, b_1, b_2..., and b_n are the coefficients for each feature, and $x_1, x_2 \ldots, x_n$ are the input features.

- **Cost Function:** When using a linear regression technique, the purpose is to reduce the difference between the projected values and the actual values in the training set. A cost function, commonly the mean squared error (MSE), which calculates the average squared difference between the anticipated and actual values, is minimized to achieve this. Finding the best-fitting line that reduces the total prediction error is made easier by minimizing the cost function **[23]**.
- **Gradient Descent:** To determine the ideal coefficient values in linear regression, gradient descent is a frequent optimization approach. The cost function's generated gradients are used to iteratively change the coefficients. The algorithm converges to the minimum of the cost function by moving in the direction of the steepest descent **[24]**.

A number of important assumptions are made by linear regression, including linearity (the assumption that the relationship between the features and the target variable is linear),

independence of errors (the assumption that the errors or residuals are uncorrelated), and homoscedasticity (the assumption that the errors have a constant variance). These suppositions can be cracked, which may have an impact on the model's accuracy and dependability.

B. Logistic Regression (LR)

It is a statistics and machine learning approach applied to binary classification situations where the target variable has two potential outcomes or classes. Despite its name, classification rather than regression is the main use of logistic regression **[25]**.

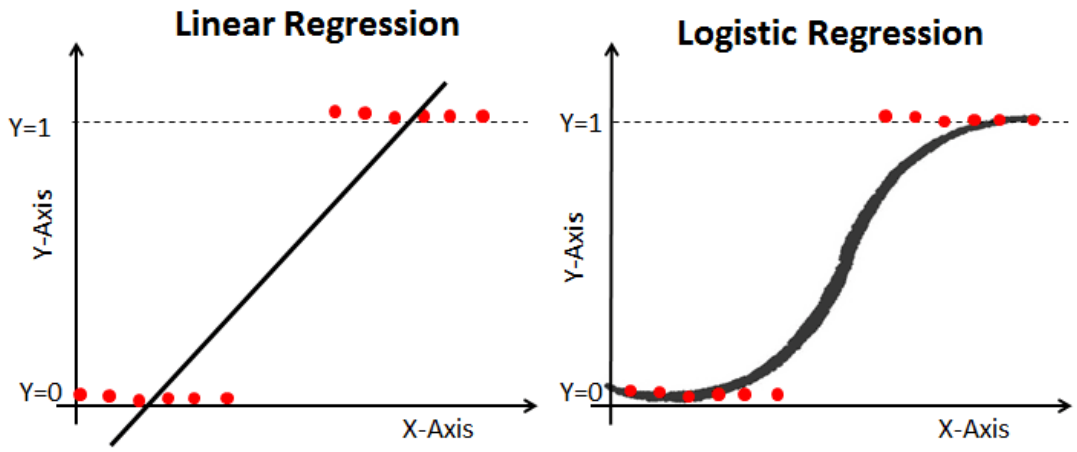


Figure 1. Linear Regression Vs Logistic Regression *(Credits to: DataCamp)*

Mathematically, Logistic Regression (LR) can also be defined as follows:

$$h_\theta(x) = \sigma(\theta^T * x) \quad (3)$$

Where, $h_\theta(x)$ denotes the anticipated probability that the input x falls into the positive category. The logistic or sigmoid function is represented by the notation:

$$\sigma(z) = 1 / (1 + e^{-z}) \quad (4)$$

Any real integer z can be translated by the sigmoid function to a number between *0* and *1*.

Moreover, x is the input feature vector, and θ is a vector of coefficients or weights assigned to each input feature. The θ vector's transposition is shown by the symbol θ^T.

Estimating the ideal values for the parameter vector θ that best fit the training data is the aim of logistic regression. Through the use of additional optimization techniques or maximum likelihood estimation, these parameters are discovered.

Then there is a decision boundary, which is linear in the feature space and is a threshold probability value, often 0.5, above or below which the input is categorized as positive or negative.

In addition, a cost function is used to calculate the difference between the true class labels and the anticipated probabilities in order to estimate the model parameters. The cross-entropy loss, commonly known as the negative log-likelihood, is the most widely used cost function for logistic regression.

$$J(\theta) = -1/m * \Sigma [y*\log(h_\theta(x)) + (1-y) *\log(1-h_\theta(x))] \quad (5)$$

$h\theta(x)$ is the projected probability of the positive class, m is the number of training examples, and y represents the true class labels (*0 or 1*).

Finally, by modifying the θ settings, the cost function $J(\theta)$ is to be minimized. Usually, optimization procedures like gradient descent or other numerical optimization techniques are used for this.

C. Decision Trees (DT)

This algorithm can be applied to both classification and regression tasks. Based on input features, DT builds a model resembling a tree of decisions and possible outcomes. Decision trees are helpful for comprehending and outlining the decision-making process since they are simple to understand and interpret. They are organized as a tree-like flowchart, where each leaf node represents the projected result or class label, and each interior node reflects a judgment based on a particular input attribute. Recursive partitioning is used to form the tree structure, and at each node, the algorithm chooses the best characteristic to divide the data according to specific standards **[26]**.

Additionally, different criteria are used by decision trees to choose the best feature and split point at each internal node. Information gain (entropy) and Gini impurity are two often used criteria. These standards assess the homogeneity or impureness of the target variable within the division's subsets. Finding splits that maximize class separation or decrease overall impurity within subsets is the objective **[27]**.

Not only that but, DT chooses the appropriate feature to split the data depending on the specified splitting criteria at each internal node. For example, greedy algorithms like CART (Classification and Regression Trees), which choose the optimal split at each step, use a different strategy for feature selection than other algorithms **[28]**.

Overall, this algorithm works better when the features of the data are distinct and easy to understand, and when it is possible to describe the relationships between the features and the target variable as hierarchical judgments. Besides, decision trees can be merged to create

ensemble models, such as gradient boosting and random forests, which enhance the performance of prediction and generalization **[29]**.

D. Random Forest (RF)

This classifier is considered an ensemble learning technique that uses several decision trees to produce predictions. A portion of the training data and a portion of the input features are used to independently construct each decision tree in the random forest. Combining all of the separate decision trees' projections yields the ultimate conclusion **[30]**.

Mathematically Random Forest algorithm can be described as:

Assuming that we have a training dataset with N samples and M input features. Each sample is represented as:

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{im}) \qquad (6)$$

where i ranges from 1 to N.

Bootstrap aggregation, often known as bagging, is a method that Random Forest uses to produce a variety of subsets of the training data for each decision tree. A bootstrap sample is produced for each decision tree in the Random Forest by randomly selecting N samples with replacements from the initial training data. A new training set with N samples is produced as a result **[31]**.

Random Forest additionally randomly chooses a subset of the input features for each decision tree in addition to sampling the training data. Only a portion of the features $(m << M)$ are taken into account for splitting at each node of the decision tree. Most frequently, a tuning parameter or heuristic is used to determine the value of m.

The Random Forest algorithm creates a decision tree for every choice using the bootstrap sample and the randomly chosen feature subset. Recursively dividing the data according to the chosen feature and a splitting criterion, such as Gini impurity or information gain, creates the decision tree. Each decision tree in the Random Forest individually generates a forecast for a fresh input sample in order to make predictions. Predictions for classification problems are typically made based on majority voting, in which the final prediction belongs to the class that obtains the most votes from the decision trees. Other methods, including taking the mean or median of the individual decision tree predictions, can be used to average or combine the predictions for regression problems **[32]**.

Combining all of the decision trees' predictions yields the Random Forest's ultimate prediction. To increase the model's overall accuracy and robustness, this aggregation method integrates the predictions from many decision trees **[33]**.

E. Support Vector Machines (SVM)

It is an effective supervised learning technique used for regression and classification tasks. SVM seeks to identify an ideal hyperplane that bestows the greatest margin of separation between the data points of various classes. In order to get the ideal hyperplane, an optimization problem must be created and solved in the mathematical description of SVM **[34]**.

Mathematically, SVM can be defined as follows:

Assuming that we have the same training dataset with N samples also for the feature vector as considered for RF. In the case of SVM, we have each sample has an associated class label y_i, where $y_i \in \{-1, +1\}$ for binary classification.

Finding the best hyperplane to divide the data points into distinct classes is the goal of SVM. The equation: defines the hyperplane.

$$w^T * x + b = 0 \qquad (7)$$

Where, w is the input feature vector, b is the bias term, and x is the weight vector perpendicular to the hyperplane.

The margin, or the distance between the hyperplane and the closest data points of each class, is what the SVM seeks to maximize. The support vectors, or data points nearest to the decision boundary, are what decide the margin.

In order to identify the ideal hyperplane, SVM formulates an optimization problem. The issue can be stated as follows given data that can be linearly separated:

Reduce:

$$||w||^2/2 \qquad (8)$$

Under the condition that:

$$y_i(w^T * x_i + b) \geq 1 \text{ for all } I \qquad (9)$$

Where $||w||^2$ denotes the regularization term, which seeks to reduce the weight vector w's normalized value. All data points are accurately categorized with a margin of at least one thanks to the inequality constraint.

It is typical in practice to have data that cannot be separated linearly. By including slack variables ξ_i, which gauge the level of misclassification, Soft Margin SVM allows for some misclassifications. The optimization challenge is changed to:

Reduce:

$$||w||^2/2 + C\Sigma\xi_i \qquad (10)$$

Under the condition that:

$$y_i(w^T * x_i + b) \geq 1 - \xi_i \quad \text{with } \xi_i \geq 0 \text{ for all } I \qquad (11)$$

The regularization parameter *C* in this case regulates the trade-off between increasing the margin and tolerating occasional misclassifications. While a lower *C* value encourages more misclassifications, a higher C value prioritizes strict classification.

A convex quadratic programming problem describes the SVM optimization issue. It can be solved using a variety of optimization algorithms, including Sequential Minimal Optimization (SMO) and Quadratic Programming (QP), to get the ideal values of w and *b*.

By utilizing the kernel approach, SVM can be expanded to handle nonlinearly separable data. The data can be separated in the higher-dimensional feature space created by the kernel function, which maps the original feature space. The linear, polynomial, Gaussian (RBF), and sigmoid kernels are typical kernel functions **[35]**.

F. Multilayer Perceptron (MLP)

It is a popular type of Artificial Neural Networks models (ANN), used for recognition, approximation, classification, and prediction tasks. It is composed of interconnected nodes that are called neurons in three types of layers: input layer, hidden layer, and output layer. The input layer presents the input signal to the process. A random number of hidden layers are located between the input and output layers, the true values of their nodes are unknown in the training dataset, for this reason, they are called “hidden”. The requested task accomplished by the MLP

algorithm is performed in the output layer. Each neuron is connected to the preceding neuron as well as the following one and the connections between them are indicated by a weight. The latter is computed based on a learning algorithm **[36]**.

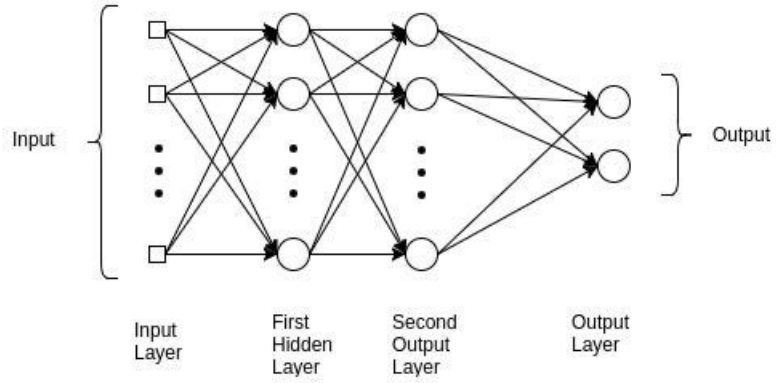


Figure 2. MultiLayer Perceptron Layers *(Credits to: Analytics Vidhya)*

Multilayer perceptron (MLP) is flexible and can be used for a variety of tasks, such as pattern recognition, regression, and classification. MLPs have been extensively employed in a variety of domains, including computer vision, natural language processing, and recommendation systems due to their capacity to learn complicated representations **[37]**.

G. Naive Bayes

The conditional probability of a class given the observed features is calculated by this probabilistic machine learning technique, which is based on Bayes' theorem. It makes the "naive" assumption, which states that the attributes are conditionally independent of one another. Naive Bayes frequently performs well in practice and is particularly successful for text categorization and other high-dimensional, categorical data applications, despite this oversimplifying assumption **[38]**.

The Bayes theorem is an essential component of probability theory and serves as the foundation for the Naive Bayes algorithm. According to Bayes' Theorem, the probability of an event *A* given an event *B* is equal to the probability of an event *B* given an event *A* multiplied by the previous probability of event *A* and divided by the prior probability of event *B*:

$$P(A|B) = (P(B|A) * P(A)) / P(B) \qquad (12)$$

Event *A* stands for the class label in the context of Naive Bayes, while event *B* stands for the observed features or attributes.

The "naive" assumption behind Naive Bayes is that, given the class label, the features are conditionally independent. This implies that the existence or absence of one feature has no bearing on the existence or nonexistence of other features. Naive Bayes is computationally efficient and can still perform well, despite the fact that this assumption is frequently broken in real-world data.

Given the observed features, Naive Bayes calculates the posterior probabilities for each class and assigns the input instance to the class with the highest probability. The most likely class for the given set of features is the one with the highest posterior probability. Using Bayes' theorem, the posterior probability is calculated:

$$P(y|X) = (P(X|y) * P(y)) / P(X) \qquad (13)$$

In this instance, $P(y|X)$ is the posterior probability of class *y* given the observed features *X*. While *y* stands for the class title, and *X* for the observed features.

For $P(X|y)$, it is the probability of witnessing the characteristics *X* given class *y*,

P(y) is the prior probability of class *y*, and *P(X)* is the probability of observing the features *X*.

Naive Bayes employs training data to calculate the probabilities *P(X|y)* and prior probabilities *P(y)*. The fraction of examples in the training set that belong to each class is used to calculate the prior probability *P(y)*. Based on the observed frequencies or probabilities of the features given each class label, the likelihood *P(X|y)* is computed. Depending on the type of Naive Bayes algorithm—Gaussian Naive Bayes, Multinomial Naive Bayes, or Bernoulli Naive Bayes—the precise estimation technique changes.

Laplace smoothing or additive smoothing are two examples of smoothing techniques that can be used to address instances where a specific feature value has not been observed in the training data for a given class. These methods guarantee that the model can make predictions for unobserved feature values and prevent the occurrence of zero probabilities **[39]**.

1.2.2.2 SEMI-SUPERVISED LEARNING

It is a machine learning paradigm that falls somewhere between supervised learning and unsupervised learning. In semi-supervised learning, instances that are both labeled and unlabeled are included in the training dataset. The basic goal of semi-supervised learning is to take advantage of the information found in both labeled and unlabeled data to speed up learning and increase model performance **[40]**.

Semi-supervised learning has been developed using a variety of algorithms and methods. Self-training, co-training, multi-view learning, and generative models like Gaussian Mixture Models (GMM) and Expectation-Maximization (EM) algorithms are some of the more well-

liked techniques. These techniques take advantage of the connections between labeled and unlabeled data to boost model performance **[41]**.

In this type of learning, the model first selects information from the small number of labeled instances, identifying patterns and relationships in the data. It then applies this learned information to estimate and predict on the wider range of unlabeled data. The fundamental premise is that the unlabeled data contains the same structure and patterns as the labeled examples, which helps the model gain more comprehension and prediction power.

By addressing the issues related to the lack of labeled data, this method allows the model to gain from a larger dataset. Semi-supervised learning is used in many fields, such as computer vision, natural language processing, and bioinformatics. It provides an appropriate solution in situations when getting access to massive labeled datasets is a challenge.

1.2.2.3 UNSUPERVISED LEARNING

It is a subset of machine learning in which the algorithm discovers correlations, structures, or patterns from unlabeled data. There are no predetermined output labels or target variables attached to the data, unlike supervised learning. Without any explicit instructions, the algorithm seeks to identify innate patterns or groups within the data **[43]**. The main traits of unsupervised learning algorithms are as follows:

A. Clustering

Based on their inherent patterns or similarities, this technique is about combining similar data elements. Since it is an unsupervised learning technique, in this case, it doesn't rely on labeled data or predetermined classes **[44]**. Finding hidden patterns or structures in a dataset is the aim of clustering, which divides the data into identifiable groups, or clusters. Data points

inside a cluster are more similar to one another than points in other clusters since the clusters are established based on the similarity or proximity of the data points. There are numerous clustering algorithms, each with unique advantages, restrictions, and presumptions. K-means clustering, for example, is a well-liked clustering algorithm **[45]**.

Example: K-Means Clustering

The objective of this popular clustering algorithm is to divide a given dataset into k different clusters. Based on the separations between the data points and the cluster centroids, it is an iterative technique that optimizes the clustering of data points by different steps: (i) The k-means algorithm selects k clusters and initializes each cluster's centroids at random. The set of k randomly chosen data points can be used to initialize the centroids. (ii) For each data point, it computes the distance between the point and each centroid. It assigns then the data point to the cluster represented by the nearest centroid. (iii) After that, by taking the mean of the data points given to each cluster, it generates the new centroids for each cluster *(The average location of all the cluster's data points is known as the centroid)*. (iv) Up until convergence, the last 2 steps are then repeated. When the centroids stop fluctuating significantly or when the allotted number of iterations has been reached, convergence takes place. (v) For the final clustering, the data points will be divided into k clusters depending on the final positions of the centroids once the algorithm converges **[46]**.

The k-means method depends on the choice of the cluster size, *k*. The quality of the clustering can be assessed using a variety of techniques to find the best value for k, including the elbow approach and silhouette analysis **[47]**.

B. Dimensionality Reduction

It is an essential machine learning technique to handle the difficulties posed by high-dimensional data. Datasets having many features or variables are said to be high-dimensional. While rich information can be provided by high-dimensional data, it frequently results in processing complexity, greater storage needs, and challenges with display and interpretation. By lowering the number of features while keeping crucial data, dimensionality reduction aids in overcoming these difficulties **[48]**. It has a good ability in serving different important tasks, such as the curse of dimensionality, feature extraction, preserving information, evaluation, and interpretation, etc. One of the most popular dimensionality reductions is called principal component analysis (PCA) and we define it as follows.

Example: Principal Component Analysis (PCA)

It finds the major components (orthogonal axes) in the data that capture the most variance. The dimensionality of the data can be decreased while preserving crucial information by choosing a subset of the primary components that account for the majority of the variance **[49]**.

1.2.2.4 REINFORCEMENT LEARNING

Artificial intelligence agents can learn by doing in an interactive environment thanks to the powerful paradigm of reinforcement learning. This learning strategy, which draws its inspiration from behavioral psychology, tries to maximize cumulative rewards by helping learners make wise decisions based on prior knowledge. The agent moves around its environment, acting and getting feedback in the form of benefits or costs, and then modifies its actions to get the best results. Reinforcement learning algorithms can effectively learn complicated tasks by continual exploration and exploitation, making them a key tool in autonomous systems, robotics, gaming,

and many other fields where making decisions in uncertain situations is essential **[42]**. Reinforcement Learning can be categorized into three types:

- **Policy Gradient Method:** Learn to map states to actions by directly optimizing the policy function.
- **Deep Reinforcement Learning:** It enables more complicated tasks by combining deep neural networks and reinforcement learning to handle high-dimensional state spaces.
- **Q-Learning:** It focuses on understanding the optimal action-value function, which shows the expected total benefits of a specific action in a specific condition.

1.2.3 DEEP LEARNING

A cutting-edge branch of artificial intelligence called deep learning centers on deep neural networks, which are multi-layered artificial neural networks. These networks can automatically develop hierarchical representations of data because they are built to resemble the linked neurons in the human brain. Deep learning has transformed a number of fields, including computer vision, natural language processing, and speech recognition. It has made incredible strides in areas like voice synthesis, language translation, and picture categorization. Deep learning has achieved enormous success due to its ability to analyze massive volumes of data and automatically identify complicated patterns, making it an effective tool for tackling difficult and complex challenges in the current technological environment **[50]**.

The fundamentals principles of Deep Learning can be described as follows:

- **Neural Networks:** They are modeled after the structure and operations of the human brain and made up of layered networks of connected nodes. Data is received by an input layer, processed by hidden layers, and output layer generates the finished product.

Weights are assigned to each link between nodes and are changed throughout training. Activation functions are used by neural networks to introduce non-linearities, which enables them to discover intricate patterns and relationships within data. During training, input data is displayed, the output is computed, compared to the intended outcome, and weights are adjusted using methods like backpropagation. Because of their adaptability, neural networks can be used for a wide range of applications, such as natural language processing, picture and speech recognition, and more.

- **Deep Neural Networks (DNNs):** With many hidden layers positioned between the input and output layers, Deep Neural Networks (DNNs) are a sophisticated type of artificial neural network. The word "deep" describes the depth to which neural networks can learn complex, hierarchical data representations. In a DNN, nodes in each layer are connected to one another, and the network uses non-linear activation functions to recognize intricate patterns.

- **Activation Functions:** In order for a neural network to learn and express increasingly complicated relationships in the data, its neurons employ activation functions to create non-linearities.

- **Backpropagation:** It is a technique used in deep learning training that involves calculating errors and propagating them backward through the network to modify the weights and biases.

- **Convolutional Neural Networks (CNNs):** they are a particular form of neural network developed to analyze data that simulates a grid, especially images and videos. Because

CNNs can automatically learn hierarchical representations of visual characteristics, they have shown to be highly efficient in computer vision applications.

- **Reccurent Neural Networks (RNNs):** They are a type of neural network that are most appropriate for tasks involving sequences or patterns over time because they are created for processing sequential data. RNNs can retain a recollection of previous inputs in the sequence because they feature connections that create directed cycles, unlike standard feedforward neural networks.

- **Transfer Learning:** It uses previously trained models on identical tasks to enhance efficiency and minimize the requirement for large volumes of labeled data.

1.3 Machine Learning Applications

Artificial intelligence (AI) has numerous uses in a variety of industries, as we already mentioned in the introduction of this chapter. Moreover, as technology develops, so does its potential. In this subsection, we are presenting further significant uses of AI besides NLP, Machine Learning, Deep Learning and Computer Vision:

- **Healthcare:** Medical image analysis, disease diagnosis, medication discovery, individualized treatment planning, and virtual health assistants are just a few of the healthcare-related jobs that AI is used for **[51]**.
- **Climate Change and Sustainability:** AI is employed to examine sizable datasets in order to comprehend climatic patterns, forecast natural disasters, utilize energy more effectively, and enhance environmental preservation efforts **[52]**.

- **Finance:** AI is used in the financial industry for fraud detection, algorithmic trading, credit risk assessment, chatbots for customer care, and customized financial advising **[53]**.
- **Education:** Adaptive learning platforms that create personalized educational content and assessment for individual students are made possible by AI **[54]**.
- **Autonomous Systems:** AI enables autonomous vehicles such as self-driving cars, drones, and unmanned aerial vehicles to navigate and make choices without the need for human interaction **[55]**.
- **Virtual assistants:** AI-powered virtual assistants like Siri, Alexa, and Google Assistant can comprehend natural language requests and carry out activities including reminding users of appointments, providing information, and managing smart home devices cybersecurity **[56]**.

Furthermore, AI improves cybersecurity by spotting and thwarting online threats, examining network activity, spotting anomalies, and automating incident response. In this thesis, we are focusing on AI for cybersecurity and its potential in improving this field **[57]**.

1.4 Advantages and Drawbacks of Machine Learning

Artificial intelligence (AI) is a cutting-edge technology that has the potential to revolutionize a number of sectors and enhance human existence. It promises to automate processes, improve decision-making, and boost overall effectiveness. Nevertheless, just like any potent technology, AI has benefits and risks that need to be taken into account. In this subsection, we'll look at the advantages and difficulties that AI presents as shown in the following table.

Table 1. Presentation of the AI strength and weak points

AI Upsides	AI Downsides
Automation and Efficiency: AI makes it possible to automate labor-intensive, repetitive processes, which increases productivity and efficiency. It has remarkable processing speed and can handle massive amounts of data.	**Security and privacy issues:** AI systems are susceptible to adversarial attacks, in which criminals take advantage of flaws in the technology. In addition, privacy problems about personal information are raised by AI's capacity to process and analyze enormous amounts of data.
Data Analysis: AI systems are excellent at analyzing huge amounts of data, finding patterns, trends, and connections that people might not always see.	**Availability of Data:** AI significantly depends on data, and when that data is insufficient or of low quality, the effectiveness of AI systems may be affected.
Improved Accuracy: AI-powered systems can do jobs more accurately and consistently than human counterparts.	**Issues with Fairness and Bias:** The quality of AI systems depends on the data they are trained on. If the training data contains prejudices, the AI system may reinforce or amplify those biases.
Continuous Learning: Being flexible is crucial for keeping current and relevant in fast-changing contexts.	**Ethical Conundrums:** The development and application of AI create problems with ethics, including responsibility, transparency, and accountability for AI judgments
Safety and Risk Reduction: By seeing abnormalities, foreseeing problems, and taking preventative action, artificial intelligence (AI) can reduce risks and improve safety in industries including healthcare and transportation.	**Job displacement:** It results from the automation capabilities of AI, which could have an impact on particular industries and job roles. While there may be new opportunities, there may also be temporary labor disruptions.

1.5 Conclusion

In summary, AI offers tremendous benefits for automating processes, evaluating data, and increasing effectiveness in a variety of fields. But it also presents issues with prejudice, job loss, security, and ethical concerns. A key component of ethical AI development and implementation is striking a balance between utilizing the potential advantages of AI and resolving its disadvantages. Due to AI capacity to handle some of the most critical issues confronting

conventional security methods, it is applied in cybersecurity, which is the most critical and sensitive area in our days. Thus, in order to successfully identify, prevent, and respond to assaults, advanced technologies are required due to the complexity and perpetual evolution of cyber threats, AI is used so frequently in cybersecurity.

Chapter II

Machine Learning for Cybersecurity

2.1 Introduction

Cybersecurity threats refer to malicious activities or incidents that pose risks to computer systems, networks, data, and digital infrastructure. These risks keep evolving as technology advances **[58]**. They are constantly challenging organizations and individuals in maintaining the security of their digital assets, As depicted in Figure 3.

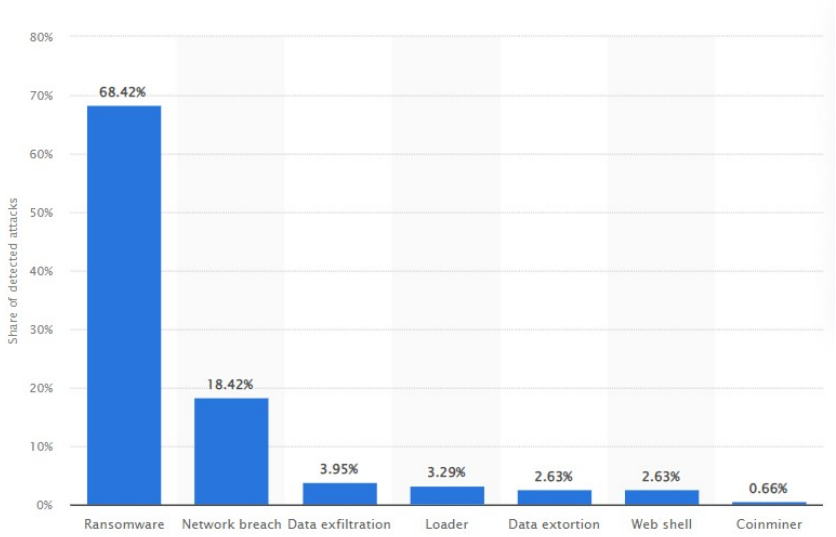


Figure 3. Distribution of detected cyberattacks worldwide in 2022, by type (Credits to: Statista.com)

Most common cybersecurity threats can be categorized as follows:

- **Ransomware:** It is a type of malware that encrypts data or locks users out of their systems, demanding a ransom in exchange for restoring access. It has become a significant threat, causing financial losses, operational disruptions, and data breaches **[59]**.

- **Malware:** It includes viruses, worms, Trojans, ransomware, spyware, and adware. Malware infects systems and can steal or corrupt data, disrupt operations, or provide unauthorized access to sensitive information **[60]**.
- **Data Breaches:** They involve unauthorized access to sensitive or confidential information, resulting in its exposure, theft, or compromise. Breaches can occur due to vulnerabilities, weak access controls, insider actions, or external attacks, leading to financial, reputational, and legal consequences **[61]**.
- **Phishing:** These types of attacks consist of fraudulent emails, messages, or websites that impersonate legitimate entities to trick users into revealing sensitive information like passwords, financial data, or personal details. Phishing is often used for identity theft, financial fraud, or gaining unauthorized access to systems **[62]**.
- **Social Engineering:** It exploits human psychology to manipulate individuals into revealing sensitive information or performing actions that compromise security. Techniques include pretexting, baiting, quid pro quo, and tailgating. Social engineering attacks target the weakest link in security: humans **[63]**.
- **Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks:** DoS and DDoS attacks overwhelm target systems, networks, or services with excessive traffic or requests, rendering them unavailable to legitimate users. These attacks disrupt operations, cause financial losses, and can be used as a diversion for other malicious activities **[64]**.
- **Insider Threats:** Insider threats arise from individuals within an organization who misuse their authorized access privileges. This includes employees, contractors, or partners intentionally or inadvertently causing harm, such as stealing sensitive data, introducing malware, or compromising systems **[65]**.

- **Advanced Persistent Threats (APTs):** APTs are long-term cyber-attacks that specifically target high-value entities like governments, corporations, or organizations. They involve persistent unauthorized access, stealthy infiltration, and data exfiltration, often performed by skilled threat actors **[66]**.
- **Zero-Day Exploits:** They are known to target vulnerabilities in software or systems that are unknown to the software vendor or have no available patches or fixes. Cybercriminals exploit these vulnerabilities before they are discovered and patched, making them highly valuable and dangerous **[67]**.
- **IoT Vulnerabilities:** The growth of the Internet of Things (IoT) resulted in new security challenges. Insecurely designed IoT devices can be compromised, leading to unauthorized access to networks, data breaches, or exploitation of botnet attacks **[68]**.

It is interesting that this technology revolution has produced a number of benefits, some of which are straightforward, like facilitating interpersonal connection, and others that are more complex, like managing the daily operations of sizable e-commerce enterprises. However, its negative aspects may be more severe than people realize. The loss of online secrecy and privacy is among the most significant concerns **[69]**.

Artificial intelligence (AI) is proven to be a game-changer in cybersecurity, altering how businesses protect themselves from online attacks. AI-powered cybersecurity systems can identify and stop a variety of unwanted actions, including complex attacks that conventional security measures may miss, thanks to their superior capabilities. It is able to spot unusual trends, indicate potential problems, and act quickly to reduce risks by processing vast amounts of data in real-time. Furthermore, it is an effective weapon in the ongoing fight against cyberattacks thanks to its capacity to continuously learn from previous instances and modify its defenses **[70]**.

2.2 The Importance of Intrusion Detection

Various cybersecurity methods are described in this subsection, including intrusion detection tasks. An overview and the importance of the latter over all the existing techniques is also discussed.

2.2.1 Security Protection Techniques

Based on what was referred to previously in the introduction, security protection methods are a topic that needs to be seriously taken into consideration. To fill this void, researchers put forward many practices, technologies, and strategies designed to safeguard computer systems, networks, data, and digital assets from unauthorized access, threats, and vulnerabilities. These methods aim to protect the confidentiality, integrity, and availability of information, ensuring the overall security of an organization's digital environment **[71]**.

The main purpose of security protection methods is to minimize the risk of security breaches, data loss, unauthorized access, and other cyber threats. By combining preventive, detective, and remedial measures, businesses may build a strong security posture and protect against a variety of attacks **[72]**. The most commonly used and well-known security protection strategies are presented in this subsubsection:

a. **Encryption techniques:** They are the process of transforming data into an unreadable format using cryptographic algorithms. These methods protect sensitive information from unauthorized disclosure by rendering it unusable without the corresponding decryption key **[73]**. Encryption can be applied to data at rest (stored data) and data in transit (data being transmitted over networks).

b. **Access Control:** These methods involve implementing robust authentication mechanisms (e.g., passwords, biometrics), role-based access control (RBAC) policies, and least privilege principles to restrict access based on user roles and permissions. They ensure that only authorized individuals or systems have access to specific resources, systems, or data **[74]**.

c. **Perimeter Defense:** Perimeter defense requires implementing measures in order to protect the boundary between an organization's internal network and external networks, such as the Internet. This typically includes firewalls, intrusion detection, and prevention systems (IDS/IPS), and virtual private networks (VPNs) to control network traffic, filter malicious content, and prevent unauthorized access **[75]**.

d. **Security Awareness and Training:** Educating employees and users about security best practices is crucial for creating a security-conscious culture within an organization. Security awareness and training programs help individuals understand potential risks, recognize social engineering tactics, and adopt secure behaviors, such as creating strong passwords, avoiding suspicious links, and reporting incidents promptly **[76]**.

e. **Vulnerability Management:** It is about identifying, assessing, and mitigating vulnerabilities in software, systems, and networks. This can be considered as regular scanning for vulnerabilities, patching systems and software with the latest security updates, and implementing robust vulnerability management processes **[77]**.

f. **Security Monitoring and Logging:** This is continuously monitoring systems, networks, and applications for signs of potential security breaches or anomalies. Security information and event management (SIEM) solutions, log analysis tools, and real-time monitoring help organizations identify and respond to security incidents effectively **[78]**.

g. **Intrusion Detection and Prevention:** Intrusion detection and prevention systems (IDS/IPS) monitor network traffic, system activity, and user behavior to detect and respond to potential security incidents. IDS identifies unauthorized or suspicious activities, while IPS takes automated actions to prevent or block such activities **[79]**.

Combining these security protection techniques offers a layered defense against potential dangers and weaknesses. Organizations should evaluate their unique security needs, carry out risk analyses, and develop a thorough security strategy that supports both their business objectives and industry best practices.

2.2.2 Why Intrusion Detection Systems?

Despite all the protection needed furnished by existing security protection techniques in order to ensure strong security, Intrusion Detection Systems are considered more critical than other security protection methods due to the following reasons:

- **Unknown Intrusions Detection:** IDSs are made to identify both known and unidentified threats. They employ a variety of techniques, including anomaly detection and behavior analysis, to spot odd or suspicious behaviors that could point to never-before-seen assault patterns. This is especially useful for spotting newly developing or "zero-day" threats that have not yet been recognized or fixed **[80]**.
- **Real-time detection:** In real time or very close to real time, intrusion detection systems (IDS) can spot and warn about potential security breaches. Organizations may respond swiftly and minimize potential damage by discovering breaches early, minimizing the impact on their systems and data **[81]**.

- **Proactivity:** Intrusion detection adopts a proactive strategy by continually monitoring for and recognizing potential intrusions, in contrast to traditional security solutions that largely concentrate on preventive measures. This makes it possible for organizations to react fast and take the necessary steps to stop or lessen the effects of assaults **[82]**.
- **Comprehensive Coverage:** IDSs give users a comprehensive picture of the network environment and have the ability to keep an eye on a wide range of activities, such as illegal access attempts, suspicious network activity, vulnerabilities in the system, and unusual behavior patterns. This extensive coverage aids in the identification of numerous security concerns and attack methods **[83]**.
- **Adaptability:** Intrusion detection systems can adapt and evolve to address new and emerging threats. They can be updated with the latest threat intelligence, security patches, and detection rules to stay effective against evolving attack techniques. This adaptability ensures that organizations can maintain a strong security posture even as threats continue to evolve **[84]**.
- **Compliance Requirements:** Many industry regulations and compliance frameworks, such as the Payment Card Industry Data Security Standard (PCI DSS) or the Health Insurance Portability and Accountability Act (HIPAA), require organizations to implement intrusion detection and monitoring as part of their security measures. Compliance with these requirements is crucial for organizations to avoid legal and financial penalties **[85]**.

In summary, IDSs are crucial elements of a thorough cybersecurity plan. They offer early threat detection, network visibility, help for compliance, and useful data for forensic investigation and incident response.

2.3 Intrusion Detection Systems Fundamentals

This section examines several fundamentals in order to comprehend the principles of IDS.

2.3.1 Definition and Purpose of Intrusion Detection Systems

In order to identify and react to illegal or malicious activity within computer networks or systems, Intrusion Detection Systems are security tools and technologies designed to monitor and analyze network traffic, system logs, and user behavior. An IDS seeks to spot probable security breaches or intrusion attempts, notify security staff, and promote the use of the proper measures to safeguard the network and reduce risks **[86]**.

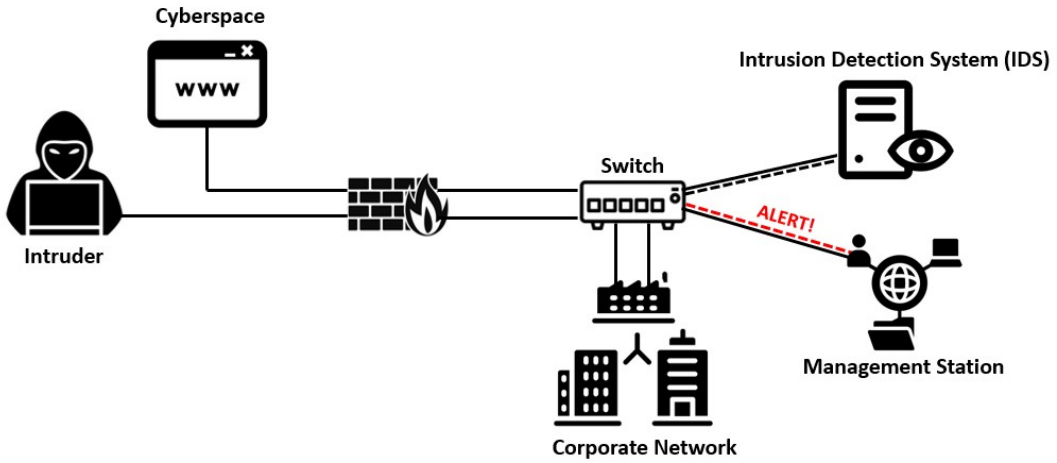


Figure 4. Intrusion Detection System (IDS) Architecture

The purpose of Intrusion Detection Systems lies in providing an additional layer of defense against cyber threats by detecting Intrusions based on monitoring network traffic, system logs, and user activities to identify signs of unauthorized access, malicious behavior, or unusual patterns that may indicate a security breach or intrusion attempt. An IDS can identify known attack signatures, anomalies, or deviations from typical behavior by examining the data it has

gathered. In addition, it produces alerts or notifications to notify security administrators or a Security Operations Center (SOC) when it finds a suspicious or harmful activity. The notifications include information about the observed event, its level of severity, and suggested incident response measures. Security staff can respond quickly to possible threats thanks to prompt alerts **[87]**. Generally, these systems are essential for incident response because they give useful details and context for security incidents that are noticed. Security personnel can use the information provided by an IDS to look into the incident, analyze it, and determine the extent and effects of the intrusion. They can then start the proper reaction procedures to limit, eliminate, and recover from the incident. Furthermore, an IDS helps the organizations in finding security flaws, vulnerabilities, or misconfigurations that attackers could exploit by continuously monitoring network traffic and system operations **[88]**. With the use of this knowledge, companies may proactively close security gaps, fortify network defenses, and enhance their security posture. Finally, the IDSs are known for the generation of logs and audit trails of detected events, supplying important data for post-incident investigations, regulatory reporting, and compliance reporting. The gathered information can be used to track security occurrences, show compliance with security requirements, and help forensic analysis for legal or investigative purposes **[89]**.

2.3.2 Components of an Intrusion Detection System

In order to monitor, identify, and take appropriate action in response to suspected intrusions or security breaches, an intrusion detection system (IDS) typically comprises of many essential components **[90]**. Here are the typical elements present in an IDS, though exact components may differ depending on the type and implementation of the IDS:

- **Sensors or Data Sources:** Sensors are in charge of gathering information from a variety of sources, including user activity, network traffic, system logs, and application logs. These sensors gather pertinent data and produce data feeds, which the IDS examines for potential security incidents or abnormalities. While host sensors keep track of events on specific hosts, network sensors record packets or flow data.
- **Data Collection and Processing:** This component is in charge of collecting and analyzing data from the sensors. It gathers and combines the data inputs, standardizes them into a format that is widely used, and carries out preliminary processing operations such data deduplication, filtering, and formatting. Techniques including packet capture, log parsing, and data transformation may be used in the collecting and processing of data.
- **Detection Engine:** The heart of an IDS is its detection engine. To find potential security events, abnormalities, or signs of compromise, it analyzes the gathered data. To identify patterns or behaviors connected to known or unknowable dangers, the detection engine uses a variety of detection approaches, such as signature-based detection, anomaly detection, heuristics, and machine learning algorithms.
- **Database:** A collection of established rules, signatures, or patterns that stand in for well-known attack methods or malicious behaviors can be found in the database. The detection engine compares against the observed data using these rules to see if a match exists. In order to respond to new threats, the rule database is frequently updated with new signatures or rules.
- **Management Console:** For configuring, overseeing, and maintaining the IDS, the management console offers a user interface or centralized platform. Administrators can use it to set detection policies, control sensor setups, see real-time alerts and events, and

access functions for reporting and analysis. The management console enables efficient management and response to security incidents and offers a comprehensive view of the IDS deployment.

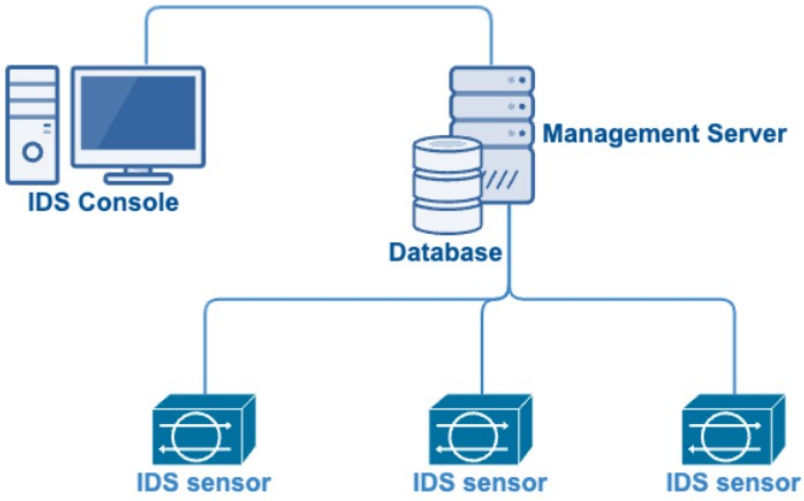


Figure 5. Components of an IDS (Credits to: TechGenix.com)

2.3.3 Ways to Detect Intrusions

To find potential security breaches or malicious activity, intrusion detection systems (IDS) use a variety of detecting techniques. The most often utilized detection methods are described in this subsection as follows:

a. Signature-Based Detection

This technique is also referred to as rule-based detection. It compares observed network traffic or system activity to a database of acknowledged threat fingerprints or patterns. Signatures are certain motifs or traits connected to well-known threats or attacks. The IDS generates an alert informing of a potential intrusion if a match is

discovered. While successful in catching known attacks, signature-based detection may have trouble catching undiscovered or zero-day threats **[91]**.

b. Anomaly-Based Detection

Anomaly-based detection entails creating a baseline of typical behavior by tracking user activity, network traffic, and system records over time. Anomalies that deviate from the predefined baseline are marked as potentially harmful or prohibited activities. Anomaly-based detection is helpful for identifying new or previously unidentified threats since it may identify both known and unknown attacks. However, if the typical behavior differs significantly, it could also produce false positives **[92]**.

c. Hybrid Detection

To improve detection accuracy and lower false positives, hybrid detection incorporates a number of detection techniques, such as signature-based, anomaly-based, and heuristic-based ones. Hybrid detection can increase an IDS's general efficacy in detecting a variety of threats by utilizing the advantages of many detection modalities **[93]**.

The table below provides a summary of anomaly-based-IDS and signature-based IDS that presents the pros and cons of each one of the latter.

Table 2. Detection ways of an IDS

Criteria	Anomaly-based IDS	Signature-based IDS
Definition	Employ deviation idea from the standard usage pattern to recognize intrusions	Employ patterns of well-known attacks to recognize intrusions
Detection Ability	Can detect known and unknown attacks	Only known attacks can be detected with extremely high-quality accuracy
Update	No	Yes
Implementation Requirements	Needs fewer computation and resources	Needs extra computation resources
The System's Characteristics	High false alarm	Low false alarm

It is important to remember that different IDSs could utilize a combination of techniques or prioritize different detection methods in order to obtain a thorough detection capability. The network environment, the anticipated threats, and the precise objectives of the IDS implementation all have a role in the choice of detection techniques.

2.4 Types of Intrusion Detection Systems

One of the several advantages of IDSs is they provide protection over different environments. The main types of IDSs can be classified into three categories.

2.4.1 Network-Based Intrusion Detection Systems (NIDS)

In order to spot suspected intrusions or malicious activity, network-based intrusion detection systems continuously monitor network traffic. To record and analyze network packets, NIDS are often installed at key nodes in the network infrastructure, including routers or switches. In order to find abnormalities or indications of known attacks, they examine the content, headers,

and behavior of network data **[94]**. For example, network intrusion attempts, port scanning, denial-of-service (DoS) attacks, unauthorized access activities, and abnormal traffic patterns are a few network-based threats that can be identified by a NIDS. The operation of NIDSs can be passive or active. When operating in active mode, they can take proactive actions like blocking or rerouting suspect traffic while operating in passive mode, they can simply monitor and analyze traffic **[95]**.

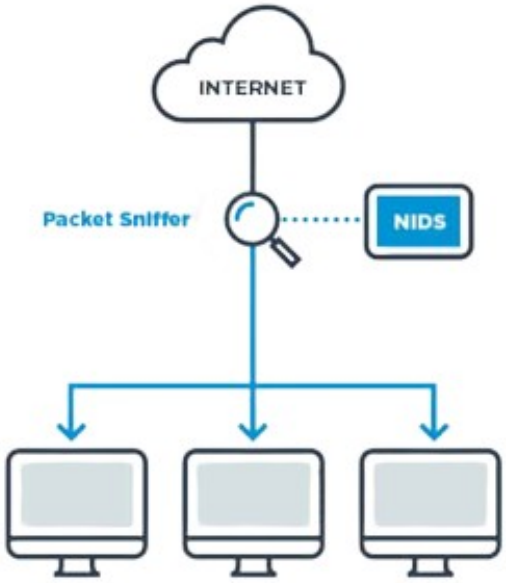


Figure 6. Network Intrusion Detection System (NIDS) (Credits to: Comparitech.com)

2.4.2 Host-Based Intrusion Detection Systems (HIDS)

Servers, workstations, and cloud instances are examples of individual hosts or endpoints that Host-Based Intrusion Detection Systems monitor within a network. The host's system logs, file integrity, registry changes, process monitoring, and user behavior are all things that HIDS keeps an eye on. To find prospective intrusions **[96]**, they match the recorded actions to known attack signatures, behavioral patterns, or predetermined rules. Besides, unauthorized access

attempts, malware infections, strange system alterations, privilege escalation, and other host-based threats can all be found using HIDS. They offer a more detailed understanding of the host's security posture and have the ability to identify assaults that may not be obvious at the network level **[97]**.

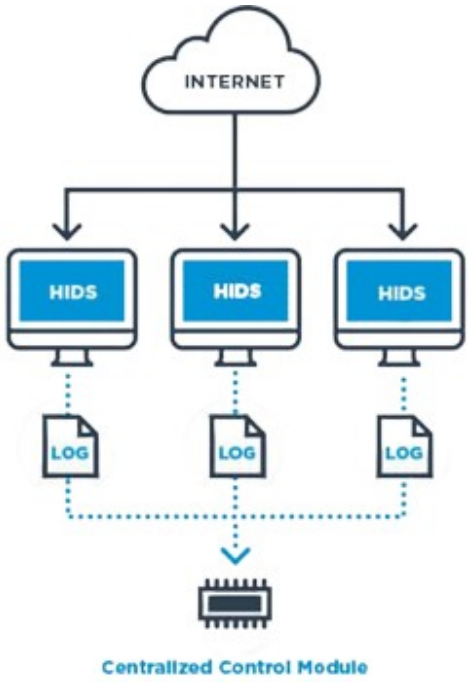


Figure 7. Host-Based Intrusion Detection Systems (HIDS) *(Credits to: Comparitech.com)*

To provide an all-encompassing capability for intrusion detection, HIDS are frequently utilized in conjunction with NIDS. They can convey warnings to a Security Operations Center (SOC) or a centralized management console for additional investigation and action.

2.4.3 Cloud-Based Intrusion Detection Systems (Cloud-based IDS)

By identifying and minimizing potential risks and intrusions, they play a critical part in protecting cloud environments. They give companies the visibility, monitoring, and analysis

tools necessary to ensure the security and integrity of the data and applications stored in the cloud. These tools are specifically designed to make use of the special features of cloud computing **[98]**. Following are some key aspects of cloud-based IDS:

- **Architecture and Deployment:** Distributed and cloud-based IDS are deployed within the cloud infrastructure. They are built to scale dynamically and adjust to cloud environments' elastic properties. The IDS parts can be set up as containers, virtual appliances, or as a cloud service offered by the supplier.

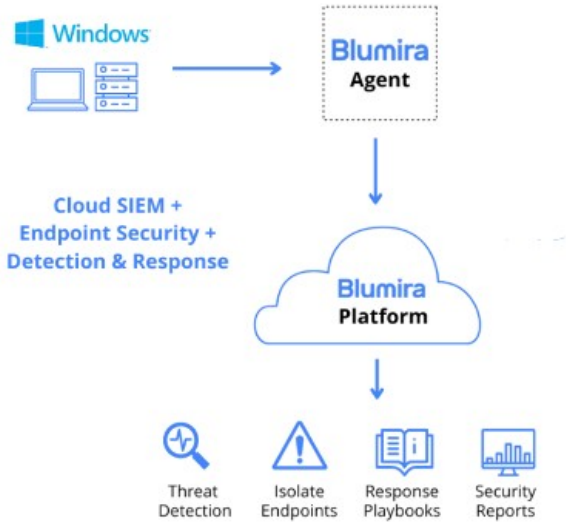


Figure 8. Example of a Cloud-Based IDS *(Credits to: Blumira.com)*

- **Service Monitoring and API:** Cloud-Based IDS keeps track of how cloud services and APIs are used. In order to find possible vulnerabilities or instances of unauthorized use of cloud services, they examine API requests, authentication events, resource provisioning, and service utilization trends.
- **Network Traffic Monitoring:** Cloud-Based IDS keeps track of network activity in the cloud environment and analyzes data flows between cloud instances, virtual networks,

and external networks. They catch and examine network packets to look for any irregularities or potential security concerns like intrusions or suspicious activity.

Despite the fact that IDSs are able to ensure protection over different environments, some limitations still need to be addressed as indicating in the following section.

2.5 Limitations of Intrusion Detection Systems

Despite being useful tools for strengthening network security, intrusion detection systems (IDS) can have some limitations. The following list of typical IDS restrictions is provided:

- **False Positives and False Negatives:** IDS can result in false positives or false negatives, indicating a security incident when there is not actually one. While false negatives can generate undiscovered assaults and potential security breaches, false positives can provide pointless alarms and resource waste **[99]**.
- **Insufficient Network Coverage:** For a variety of reasons, including network design, segmentation, or a lack of sensor deployment in crucial locations, an IDS may not have comprehensive network coverage. Since there are gaps in the coverage, intrusions may go unnoticed and expose susceptible areas of the network **[100]**.
- **Human Intervention:** An IDS just assists in identifying attacks; it does not stop or hinder them. As a result, an IDS must be a component of an all-encompassing strategy that also includes other security measures and personnel who are trained to respond correctly **[101]**.
- **Limited Contextual Information:** For example, the system may only provide a limited amount of context for events that are detected, making it difficult for security analysts to comprehend the full extent and consequences of an incursion. To fully comprehend the

attack, additional manual investigation and correlation with other security technologies could be needed **[102]**.

Organizations frequently combine security techniques and procedures, including network segmentation, security information and event management (SIEM) systems, intrusion prevention systems (IPS), and user education, to get around these restrictions. IDS's effectiveness can also be increased by continuously upgrading and adjusting them, as well as by remaining informed about new threats.

2.6 Intelligent Network Intrusion Detection Systems

Based on the significant potential that intelligent approaches can provide, the latter can be considered as one of the most appropriate solutions to address the limitations of a traditional IDS. Many researchers have opted and still opting for deep learning and shallow methods with the purpose of improving the detection and classification of network intrusions. It generally comes down to the following reasons:

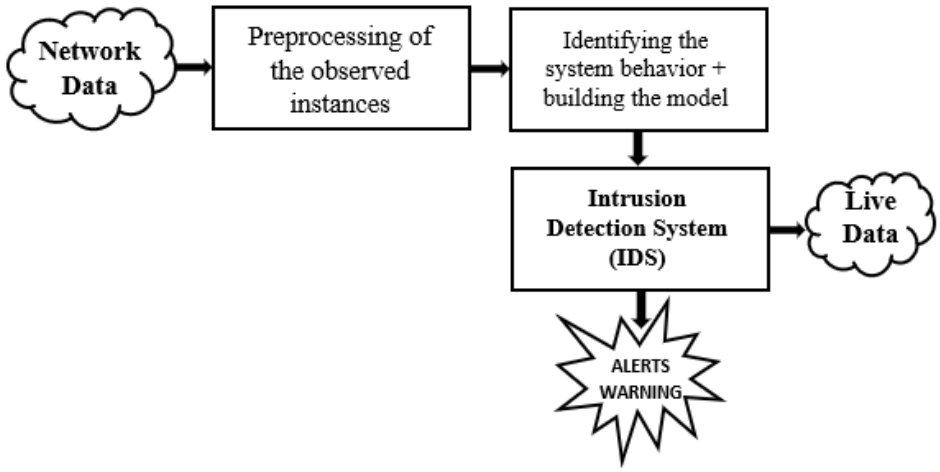


Figure 9. Machine Learning-based IDS

A. **Good Capacity to Learn from a Database**: Algorithms that use machine learning are capable of learning patterns and behaviors straight from network traffic data. Machine learning algorithms can recognize typical network behavior and spot anomalies or suspicious actions that may signify network intrusions by examining historical network data and labeled occurrences. This makes it possible for NIDS to adapt and recognize new and undiscovered dangers **[103]**.

B. **The Recognition of Complex Cyberattacks**: Traditional rule-based intrusion detection systems use pre-established rules or signatures to identify particular threats. Attackers can, however, continuously improve their methods and create fresh attack patterns that might not be covered by predefined regulations. Even if abnormal network behavior does not match known attack signatures, machine learning enables NIDS to learn complicated attack patterns and detect anomalous network behavior **[104]**.

C. **Anomaly Detection**: Based on unusual behavior, machine learning-based NIDS can identify network breaches. Machine learning algorithms can recognize deviations from the anticipated patterns by learning what defines normal behavior from labeled data. When identifying unknown or zero-day attacks, where the attack features are not known ahead of time, this is especially helpful **[105]**.

D. **Minimizing the Number of False Positives**: False positives can result from traditional rule-based NIDS when ordinary network activity is wrongly marked as an incursion. By learning from labeled data and increasing the precision of intrusion detection, Machine Learning algorithms can help in the reduction of false positives. The models can differentiate between typical patterns and abnormal or harmful patterns, resulting in more accurate identification and a reduction in pointless alarms **[106]**.

E. **Autonomy and Adaptability**: Without the need for human rule updates, machine learning-based NIDS can adapt to and learn from new data. This adaptability enables the system to change and react to new threats. The NIDS can continue to be effective at detecting new assaults because machine learning models can automatically update and retrain with new data **[107]**.

F. **Dealing with High-Dimensional and Dynamic Data:** The volume of data produced by networks makes manual analysis and rule formulation impossible. In order to handle the high amount and velocity of network data, large-scale network traffic data must be processed and analyzed in real-time. Additionally, without user involvement, machine learning enables NIDS to react to altering network conditions, evolving attack techniques, and emergent threats **[108]**.

2.7 Challenges and Considerations in Intelligent NIDSs

While machine learning-based and deep learning-based network intrusion detection systems (NIDSs) have several advantages as mentioned previously, there are a number of issues that must be taken into account. In this subsection, we present some major issues and challenges to consider while developing an intelligent NIDS:

- **The Quality and Imbalance of Data:**
    - For training, machine learning models largely rely on representative and good-quality data. Finding labeled training data for network breaches can be difficult because it frequently calls for specialized knowledge or old information. To fully understand the dynamic nature of network attacks, it is also essential to guarantee the accessibility of various and current data **[109]**.

- The proportion of harmful instances in network intrusion datasets is often low relative to the proportion of benign cases. This might cause bias in the learning process, where the model improves at seeing common occurrences but struggles to spot uncommon incursions **[110]**. To properly manage imbalanced data, special methods like oversampling, undersampling, or the use of cost-sensitive learning algorithms are needed.

- **Overfitting and Generalization:** Machine Learning models must avoid the problem of overfitting and be able to generalize well to new data. When a model learns the training examples by heart but is unable to generalize to fresh data, overfitting takes place. To reduce overfitting, regularization approaches, cross-validation, and maintaining a balance between model complexity and generalizability are crucial **[111]**.
- **Adversarial Attacks:** By creating hostile samples, hackers may try to take advantage of flaws in machine learning models. These are deliberately constructed input instances that are meant to deceive the model and avoid detection. The resilience and dependability of machine learning-based IDSs can be seriously hampered by adversarial attacks **[112]**. This risk can be reduced using strategies like adversarial training and anomaly detection systems created particularly to find adversarial data.
- **Feature Engineering:** Building a machine learning-based IDS requires carefully choosing and engineering useful elements from network traffic data. It can be difficult to choose which features to include, how to express them and cope with high-dimensional data. To extract useful characteristics, it is required to have a solid understanding of network protocols and threats as well as domain expertise **[113]**.

- **Affecting Real-Time Performance:** IDSs must identify and respond to network threats instantly and effectively since they function in real-time. For machine learning models to process network traffic quickly, they must be computationally efficient. To achieve real-time performance, optimizations like model compression, hardware acceleration, or distributed processing might be used **[114]**.

2.8 Conclusion

Integrating machine learning into NIDS enhances the system's ability to detect and respond to network intrusions, improves accuracy, reduces false positives, and provides a more adaptive and intelligent approach to network security. However, it is important to continually monitor and update machine learning models to ensure their effectiveness in detecting new and evolving threats.

Chapter III

A Powerful Ensemble Learning Approach for Improving Network Intrusion Detection System (NIDS)

3.1 Introduction

With the rapid and historical growth of internet technologies, the total amount of data created every year is vastly increasing. Therefore, the development of the network security is considered as an important task that should be taken into real consideration, as over the years it is getting more challenging. It is widely known that Intrusion Detection Systems (IDSs) are a major security measure to identify and prevent threat attacks or any unauthorized users breaching the network system and its sensitive data **[115]**. They have always been in under limelight, because of their great potential to catch different intrusion types. Besides, they provide external and internal protection, unlike many security tools and techniques that necessitate additional measures of cybersecurity and only ensure external protection **[116]**. However, in spite of its efficient capacity of being an important defensive measurement, an IDS cannot handle unknown or recent cyberattacks that are ceaselessly increasing in terms of intensity and frequency over the past years, due to the phenomenal augmentation of technological progress **[117]**. As a result, it regularly warns about false positives more than it does about real abnormal behaviors. For that reason, the performance of an IDS continually requires improvement and updates according to the current cybersecurity issues.

In order to fill this need and deal with the limitations of IDSs, researchers have been based on several techniques in which machine learning algorithms are considered the most widely used by many different authors to develop the effectiveness of this security system. In the last few years, Random Forest, Logistic Regression, K-nearest neighbors, Artificial Neural Networks, and other machine learning classifiers have been considerably applied in the cybersecurity domain, including the intrusion detection task **[118]**. This great attention is a consequence of the powerful potential of machine learning methods in identifying effortlessly specific trends and patterns by providing to machines the ability to learn and make predictions without any human intervention, unlike the IDS which necessarily requires the latter **[119]**. Additionally, they can process and evaluate multi-dimensional and various types of data, which results in a robust system with a high accuracy rate that can make better predictions and decisions. Hence, their technology keeps improving and evolving over the years. However, each machine learning classifier has its own pros and cons. Because of the diversity of the existing types of attacks on the network systems, such as Distributed Denial-of-Service (DDoS), Probing, Surfing, Ransomware attacks, etc., some classifiers may not be able to perform well on a type of attack but can be performant on another one **[120]**. Thus, it is not always sufficient to rely on a single learning approach for network security.

Numerous research studies are well carried out to shed light on ensemble learning in order to integrate the cons of each classifier and maximize the detection accuracy rate. They approved through many analyses that ensemble models are more effective and achieve high accuracy than single models **[121]**, since it is able to handle computational and statistical problems, by reducing bias and variance. This collaborative approach is based on training multiple learners and combining their strengths and outcomings. Hence, it can be considered as an appropriate solution

to boost the prediction task, which is classifying whether the traffic data is normal or abnormal. The ensemble methods can be categorized into three types:

a. **Bagging:** It is considered as an averaging method that relies on multiple models and runs each one of them on a different set of input samples, then it takes the average of the obtained predictions. It is the most appropriate ensemble technique to decrease variance. Moreover, it can handle the overfitting issue **[122]**.

b. **Boosting:** It integrates multiple weak learners, where each one of them learns from the mistakes of the previous one so that, it will result in one strong learner that can make better predictions. Besides, this technique can efficiently decrease the bias in the training set, unlike the bagging method which cannot do so **[122]**.

c. **Stacking (or blending):** This is a two-level ensemble learning technique, which requires first a training of different single learning algorithms on the training data. Then, it fuses all the prediction results of each algorithm to finally make a better prediction on new data **[122]**.

The present paper proposes five collaborative learning models for the IDS problem, aiming to categorize the network traffic and achieve optimal results. It is based on the stacked ensemble learning technique, which considers some of the most important machine learning algorithms; Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), K-nearest neighbors (KNN), Naïve Bayes (NB) and Multilayer Perception (MLP). Each model is built by combining the advantages of three algorithms, to finally come up with the best combination that reached the highest accuracy rate. The experiments are evaluated on the NSL-KDD data set that contains 43 features; however, we take into consideration the fact that not all of them are meaningful and useful in detecting intrusions. Therefore, to improve the performance of the prediction process, we have only selected 10 features, that have a direct impact on the dependent

variable. Moreover, an extensive experimental evaluation of these 5 models is conducted in this work to approve that it provides an important improvement in the detection rate. By comparison, the proposed models make better predictions than many other research results.

3.2 Related Works

Many researchers have been investigating on the Intrusion Detection System sector and proposed various methodologies intending to improve its efficiency in classifying the data traffic into normal or cyberattack categories. Over the last few years, a countless number of research studies have integrated machine learning classifiers in order to handle the classification issue of IDSs. In a very recent work, S. Ennaji et al. presented a comparative study, in which they approve the importance of machine learning classifiers in enhancing the performance of IDSs. They have been based on five classifiers, including; SVM, NB, LR, RF, and MLP, and the NSL-KDD as a data set. Their analytical study showed the advantages and disadvantages of each algorithm, which is helpful to find out what an IDS needs to update and strengthen its efficiency according to the current security problems that it encounters each year. Their experiments results indicate that even though MLP is more accurate than all other algorithms, it is a time-consuming classifier in terms of training and prediction, unlike NB and LR which have been considered as fast algorithms **[119]**. Whereas in another work by F. Yihunie et al, it was proved that RF outperforms four other classifiers: SVM, LR, Stochastic Gradient Decent (SGD), and Sequential Model using the same data set as **[119]**. In regard to time of training, SGD requires less time than RF **[123]**. To the best of our knowledge, most comparative studies using single classifiers cannot find the appropriate algorithm that works well in terms of all performance measures such as running speed, precision, recall, etc., because each classifier has its own

strengths and weaknesses. For this reason, researchers addressed this issue through several studies, relying on the ensemble technique.

Aiming to validate the importance of the ensemble method in developing the performance of IDSs, several researchers proposed different approaches based on collaborative learning and compared their obtained results with those of single learning and other similar techniques. For instance, J. Gu et al. put forward a framework for intrusion detection based on three steps consisting of splitting the data and rebuilding the original features to scale training data. Then, relying on the quality-improvement method, several base SVM learners are trained on the new data. Finally, using a nonlinear combination, the SVM classifiers are integrated to build up the intrusion detection model **[124]**. The authors achieved good results and showed how their model is more effective than single SVM and other compared models. Furthermore, in **[125]**, a new anomaly-based network IDS has been presented based on the stacked ensemble learning that has considered three powerful classifiers: RF, SVM, and autoencoder (AE). In order to prove the robustness of this contribution, the authors have evaluated their experiments through different data sets, including NSL-KDD and UNSW-NB15 using all features, but applied feature selection for Palo Alto system log data set. The latter was collected from their campus network log, which represents about 300 million records per day. Their method results in a high classification rate. It was also shown that it can handle false positive and false negative classifications.

SA Ludwig has also confirmed the great potential of the ensemble method compared to the single one. The proposed approach trained several interesting algorithms on the NSL-KDD data set, namely: Autoencoder, extreme learning machine, deep neural network, and deep belief neural network. Then, the results of each one of them are combined in order to differentiate between an attack behavior and a normal one within the network **[126]**. The author compared the

results of this neural network combination with some deep learning-related work approaches that have also used the same data set as he did. It has been proved that his method performs better than other works in terms of precision, recall, and f-measure, except for the classification accuracy.

In another work, H. Rajadurai et al. proposed a stacked ensemble model consisting a fusion of gradient descent and random forest classifiers. Their method transformed the weak learner into a strong one that can make better decisions or predictions for the test data, using the NSL-KDD data set. Besides, the obtained outcomings approved that the stacked ensemble method is better than other types of ensemble learning, such as bagging and boosting **[127]**. Not only that, but the authors also compared their model with some existing popular machine learning classifiers. Thus, it seemed that their model performs better than other ones proposed by different research studies.

In addition, it is also known that the quality of an intrusion detection system depends on the way in which the data has been trained or processed. NF. Haq et al., proposed a hybridized feature selection method to get a final significant feature set, using the first 3 types of feature selection approaches, then merging the commonly extracted features by these 3 approaches to finally create a new feature set. Afterward, they trained 3 base classifiers: NB, Bayesian Network, and J48 to design an ensemble learning model. The results proved that their proposed ensemble method is better than using only NB, Bayesian Network, or J48, in detecting anomaly behaviors within NSL-KDD **[128]**. BA. Tama et al. **[129]** also opted for hybrid feature selection to build an IDS model using two-level classifier ensembles over two different data sets: NSL-KDD and UNSW-NB15. The produced results show how their proposed approach has been important in enhancing the accuracy, precision, and specificity of both datasets. Moreover, it has

been performant in terms of training time, which is considered as a significant measurement that should be taken into account during the creation of the model, since it impacts the quality of the latter.

Another network intrusion detection system (NIDS) has been developed by S. Das et al. in **[130]**, which was also based on the ensemble approach using 4 various machine learning classifiers from different classifier categories: MLP, J48, KNN, and SVM. Each one of these four classifiers creates a model during the training phase, in order to test the final model that represents a combination of all the trained algorithms, using 10-fold cross-validation method within a reduced feature set of the NSL-KDD data set. The authors have specialized their system on detecting only DDoS attacks and their new types, which are increasing over the years. The results validated that this proposed NIDS is able to detect new DDoS attacks successfully. Additionally, they compared it to other existing systems and it seemed that it performs better than all of them **[130]**. M.S. Abirami et al. also used the feature selection method to build their proposed model, which is called Least Square Support Vector Machine IDS (LSSVM-IDS). The latter was evaluated on different data sets such as KDDCUP'99, NSL-KDD, and Kyoto 2006 by keeping the significant features and removing the redundant ones. Its accuracy achieved 95%. However, it does not deal with recent cyberattacks and it is computationally expensive. To handle this issue, the authors adopted the stacking ensemble technique and created a combination of three classifiers: RF, SVM, and, NB by considering LR as a meta-classifier, which was applied to another data set named UNSW-NB15 **[121]**.

An efficient novel technique presented by Khonde et al. that has increased the accuracy of an IDS, which relies on the ensemble method that combines RF, SVM, ANN, DT, and KNN to classify the network traffic and predict if it's a normal packet or a malicious one, using NSL-

KDD data set. What differentiates this contribution from others is the application of a new feature selection method, which had an important potential in identifying real-time traffic **[131]**. The authors approved through the discussion of their results that their ensemble method provides a high detection rate and less false positives than the single learning method.

3.3 Comparative Study

Before proposing our multi-model learning based IDS, we conducted a comparative study based on 5 machine learning classifiers, namely: MLP, RF, NB, LR and SVM over NSL-KDD dataset. This comparison helped us to show the upsides and downsides of each model used, which has leaded to an optimal selection of the convenient model for the next contribution.

The results obtained are presented in thetable below show the performance of each classification algorithm in the NSL-KDD dataset using 10 features that have an impact on the class variables for test dataset, within general 2 class; normal and attack class. It is observed that MLP outperforms over all other algorithms with an accuracy of 91.41%, precision of 80.71%, recall of 96.68% and F1-Score of 87.98%. The choice of the parameters metrics used to implement the MLP algorithm has also helped to increase the accuracy rate, since we have only defined the *'hidden_layer_sizes=(9,5,9)'* to set the number of layers with the number of nodes in each layer, and *'max_iter=400'* that denotes the maximum number of iterations.

On the second rank, the SVM algorithm has an accuracy of 86.61% and recall of 94.95%. However, it has the poorest precision rate of 70.62%. It indicates that SVM returns a portion of false positives that should be taken into account, compared to other classifiers. We only considered the *'random_state=0'* as a parameter, because firstly we only used the SVM to classify whether a network contains an attack or not. Secondly, it is computationally costing.

The LR model has been ranked third with an accuracy of 84.19%, and precision of 79.96%. But fourth in term of recall (83.02%), since NB and RF reached 86.11%, 82.85% respectively. We used all CPUs to run the LR by setting the parameter *'n_jobs=-1"* and used *'random_state=0'* as fixed to the SVM model.

At the fourth place, NB has an accuracy of 83.37%. It is third in term of recall (86.11%) which relates to a low false negative identification. However, it has a low precision rate of 73.58%, which indicates the portion of false positive identification.

The algorithm with the low performance is RF with an accuracy of 82.99% based on setting the *'criterion'* parameter to gini function, *'random_state'* and *'n_jobs'* to 1, because RF is known to be computationally expensive and these parameters will make it much faster.

Table 3. Comparison of Machine Learning algorithms using NSL-KDD dataset in % (Test results)

Algorithms	Accuracy %	Precision %	Recall %	F1-Score %
MLP	91.41	80.71	96.68	87.98
SVM	86.61	70.62	94.95	80.99
RF	82.99	76.71	82.85	79.67
LR	84.19	79.96	83.02	81.47
NB	83.37	73.58	86.11	79.67

Further, from the results provided in the figure 11 bellow, which is about time of building the model and time of prediction. NB takes less time to build (0.15 Seconds) than other models, since it only depends on the prior probability values, hence no parameters have been considered. Moreover, it has been ranked third in term of prediction time (4.89e-5 Milliseconds). The second rank is reached by RF in time of learning and prediction (16.57 Seconds and 4.59e-05 Milliseconds, respectively), since it is known to be one of the fastest classifiers. Additionally, we are testing the dataset only on a subset of features in this work. The LR model has been the less

time-consuming one in term of prediction time (4.24e-05 Milliseconds), but the third one in term of responding time (24.37 Seconds). The SVM model takes so long to be built and to predict, as it is an incremental algorithm.

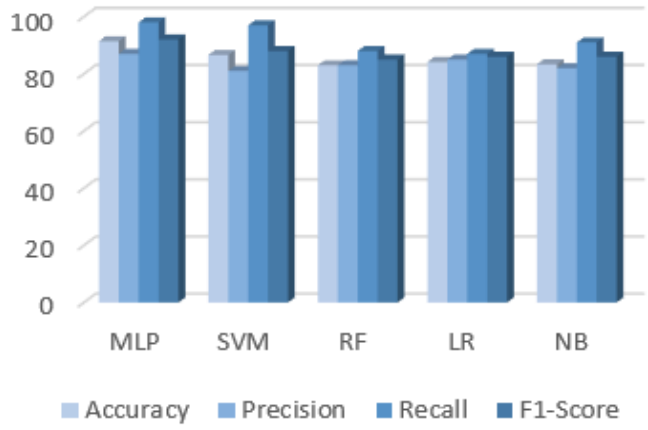


Figure 10. Comparison of Machine Learning algorithms using the most known indicators

Despite the good performance of MLP model, it requires more time than other algorithms in terms of building the model and prediction process. Thus, it is not appropriate for real-time learning/prediction, since it is a computationally costly process. However, it is approved that MLP is highly fault-tolerant and remains the most accurate one that needs some improvement to make it more effective.

Table 4. Comparison in terms of Learning and prediction time of Machine Learning algorithms using NSL-KDD data set

Algorithms	Time of Learning (Seconds)	Time of prediction (Milliseconds)
MLP	41.17	1.00e-4
SVM	167.47	5.44e-05
RF	16.57	4.59e-05
LR	24.37	4.24e-05
NB	0.15	4.89e-5

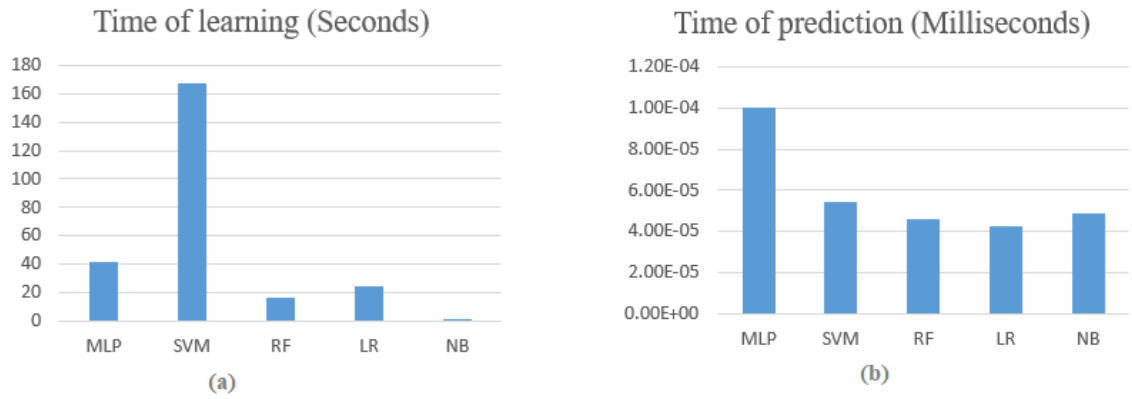


Figure 11. Time of learning (Seconds) and prediction (Milliseconds)

The experimental results show that MLP is the most accurate model for IDSs with a minimum false positive rate, using the NSL-KDD dataset for the experiments and Google Colab as an environment. Moreover, it presents the limitations of the compared models that will be taken into account in the next contribution. In the following proposition, we are dealing with the obstacles that made a classifier not efficient enough.

3.4 Proposed Model

This section depicts the data set used to evaluate the proposed models as well as the tools and methodology used in this contribution.

3.4.1 Dataset Description

As is well known, it is also the quality of the data set that plays a major role in building a powerful Intrusion Detection System (IDS). To the best of our knowledge, the NSL-KDD data set seems to be effective for our experimental results. It was suggested to deal with the limitations of its predecessor called the KDD'99 cup data set **[132]**, since it includes duplicate and redundant records. Consequently, the classifiers create biasing results. For this reason, the NSL-KDD data set comes to update this version and solves the problem of biasing by

eliminating the repeated records. Moreover, an adequate number of records are available in the train and test sets. Therefore, running the experiments on the whole set will be better, without requiring to arbitrarily select a small portion. In addition to that, from each difficult level group, the selected records are inversely proportional to the percentage of records in the previous version of the NSL-KDD data set **[132, 133]**, which results in an efficient evaluation of several classifiers. Many researchers approved that it is a performant benchmark to compare various intrusion detection models. It concludes 24 types of attacks categorized into four classes:

- **Denial of Service attack (DOS):** This type of attacks is very well known, it ruins the service by overloading the resources of the victim, which results in a denied service for individuals who have access to the service. For the DDOS attacker, he damages many internet machines, which creates a botnet and makes him recruit the systems that he cannot control. All of these actions make the attacker more powerful since it is hard to identify him or detect the dangerous sources **[134]**. There are several DOS/DDOS attacks identified by the following names: Buffer Overflow, Ping of Death, ICMP, Smurf, UDP Flood, SYN Flood, Backdoor Program, and Trojan Horse.

- **Probing:** An invasive physical attack that joins different reports used by the attacker in order to access directly to networks, destroy the crypto processor, spy on electrical activities, and extract sensitive data, etc. **[135,132]**.

- **Remote to Local User (R2L):** Such an attack targets a system or a network, it happens when the intruder sends packets to the target system that will be able to identify all the vulnerabilities the attacker could exploit **[136]**.

- **User-To-Root (U2R):** Unauthorized access to the victim's system by using a normal account and trying to breach the vulnerabilities, to get all the rights and accessibilities that the victim has **[137]**.

The training NSL-KDD data set is occupied by the known attacks, while the testing data set presents the remaining attacks **[138]**.

3.4.2 Experimental Setup

The empirical experiments in this contribution were performed using a Dell Inspiron 15 3000 Series Core i5 7th Gen - (4 GB/1 TB HDD/2 GB Graphics). The latter does not offer an appropriate GPU for our experiments; hence we have connected our machine to a free cloud-based Jupyter Notebook environment called Google Colab, short for Google Collaboratory **[149]**. It provides free GPU Tesla K80 and TPU. It is an important tool supporting research in developing Big Data and Machine learning domains. Besides, it does not need any setup and comes with preinstalled libraries needed for the experimentations in such fields. Moreover, the data files can be easily downloaded and all the Google Colab notebooks are saved to the drive. Therefore, the user can quickly access to it and share it with other users.

In order to improve the prediction time, we have edited the type of running in Google Colab and used Tesla T4 which is a professional graphic launched in September 2018 by NVIDUA. It is based on a large chip called TU104 graphics processor that has 13600 million transistors. Furthermore, it contains 320 tensor cores and 40 raytracing acceleration cores which enhance the speed of artificial intelligence applications.



Figure 12. Google Colab Logo (Credits to: Google colab)

3.4.3 Methodology

The proposed algorithm is described as follows:

Algorithm: Ensemble modeling for network IDS

Input: NSL-KDD dataset

Output: Classification of network traffic whether it is normal activity or abnormal

Step1: Loading the train and test data sets

Step2: Data preprocessing

Step3: Applying the feature selection technique

Step4: Training the classification algorithms

Step5: Building the models by applying the ensemble learning method

Step6: Testing the performance of each model

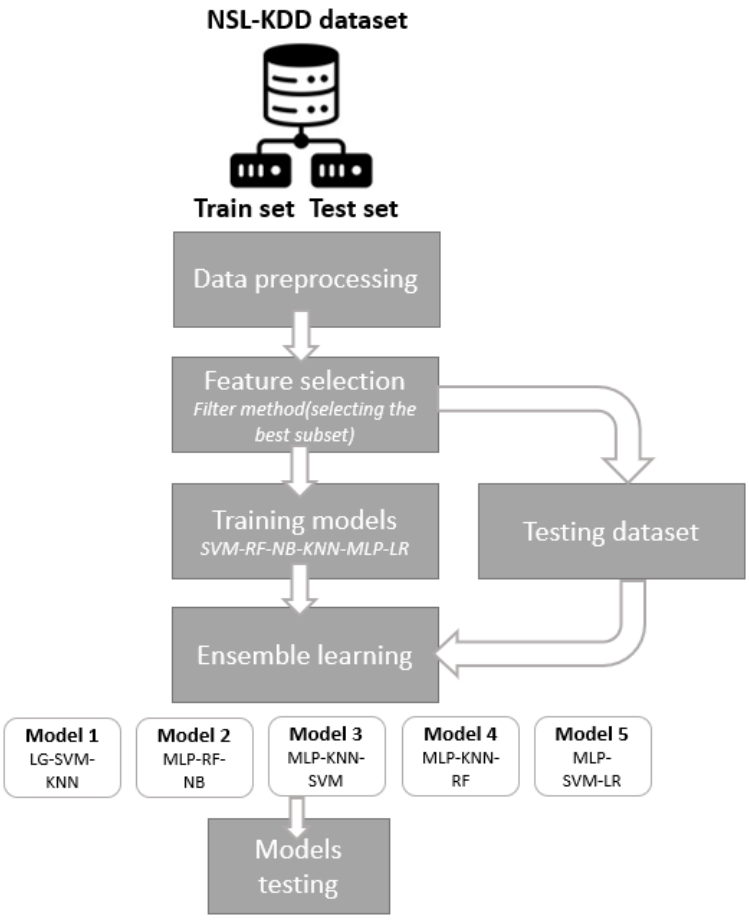


Figure 13. A conceptual framework for an Intrusion Detection System models

Figure **13** presents the design of the proposed model, which firstly consists a data analysis, which requires the mapping technique to organize the data and match the attack field to the fourth attack classes: Dos, Probe, U2R and R2L. Additionally, it is necessary to remove all the redundant features from both train and test datasets. Then, the numerical attributes should be scaled in order to eliminate the mean and unit variance. Afterwards, the categorical attributes have been encoded to perform the sampling method, which handles the imbalanced classification. Secondly, to enhance the prediction of the system, we have opted for the feature selection technique, which is an important process to select a precise and accurate subset of

features that have a direct impact on the dependent variable. In this case, the 10 most important features selected are*: 'src_bytes', 'dst_bytes', 'service', 'logged_in', 'count', 'srv_count', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate'* and *'dst_host_serror_rate'*. Therefore, the computational cost will be reduced and the performance of the classification will get better. Lastly, the ensemble learning model is evaluated for the intrusion classification task.

3.4.4 Evaluation of the Model

In order to evaluate the models built and compare them to each other, it requires calculating the accuracy rate, false positive rate, and true negative rate of each model. These evaluation metrics depend on a confusion matrix or a table of confusion. The latter is composed of predicted and actual classes, as it is shown in the figure below:

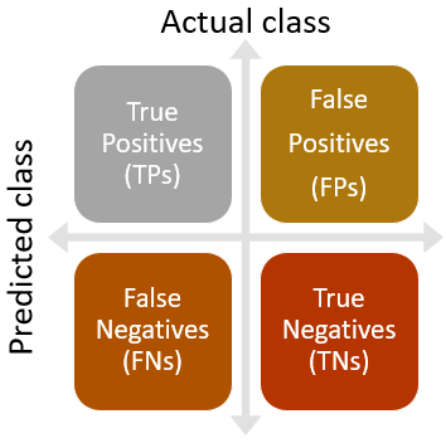


Figure 14. Truth table of the confusion matrix

The terms used in the confusion matrix are:

- ***True Positives (TP):*** It indicates the positive classes that were correctly predicted by the classifier.

- ***True Negatives (TN):*** It indicates the negative classes that were correctly predicted by the classifier.
- ***False Positives (FP):*** These are the positive classes that were incorrectly predicted by the classifier as positive.
- ***False Negatives (FN):*** These are the positive classes that were predicted by the classifier as negative.
- From these terms, we can examine the performance of the models using the following calculations:
- ***Accuracy (TPR):*** It is also called the true positive rate and it is calculated as

$$TPR = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

- ***Precision (P):*** It refers to the percentage of how often the user labeled as positive is actually positive, it is defined by the following formula

$$P = \frac{TP}{TP+FP} \qquad (2)$$

- ***Recall (R):*** It is the percentage of tuples included in the positive class that are correctly labeled as positive, and calculated as

$$R = \frac{TP}{TP+FN} \qquad (3)$$

- ***F1-score:*** It denotes the weight average of Precision and Recall

$$\text{F1-score} = 2 \times \frac{\text{precision.recall}}{\text{precision+recall}} \qquad (4)$$

3.5 Results and Discussion

The purpose of this research study is to maximize the prediction rate of an intrusion detection system based on the ensemble learning approach. Several machine learning classifiers have been used to build 5 different models that combines the advantages of 3 classifiers, with the aim of comparing them to each other and select the best combination. The 10-fold cross validation has been applied to evaluate the performance of the proposed models. Thus, the classifiers were executed 10 times in the NSL-KDD data set with selecting 10 features that have an impact on the class variables for the test dataset, within the general 2 class; normal and abnormal class, then the experimental results were averaged.

As it is shown in Table 3, the metrics considered in this work to evaluate our proposed models are Accuracy, Precision, Recall, and F1-measure. It is evident from the obtained results that the combination of MLP, RF, and NB yielded the highest accuracy of 99.66% to detect intrusions. On the second rank, MLP, KNN, and RF achieved an accuracy of 99.55%. However, it has the best recall rate of 99.54%; which relates to a low false negative identification. The ensemble learning model that integrates MLP, KNN, and SVM has ranked third with an accuracy of 99.37% and a recall of 99.48%. Then, the model considering MLP, SVM, and LR reached 99.15% as an accuracy and 99.38% as a recall. The model with the low performance was the combination of LR, SVM, and KNN as it has 99.06% in accuracy rate and 99.24% in terms of recall. Therefore, all the proposed models make great predictions since their accuracy is higher than 99%.

Table 5. Comparison with the five proposed ensemble learning models using NSL-KDD data set in % (TEST RESULTS)

Ensemble models	Accuracy	Precision	Recall	F1-measure
MLP-RF-NB	99.66	99.66	99.51	99.59
MLP-KNN-RF	99.55	99.63	99.54	99.59
MLP-KNN-SVM	99.37	99.12	99.48	99.30
MLP-SVM-LR	99.15	98.57	99.38	98.98
LR-SVM-KNN	99.06	98.53	99.24	98.89

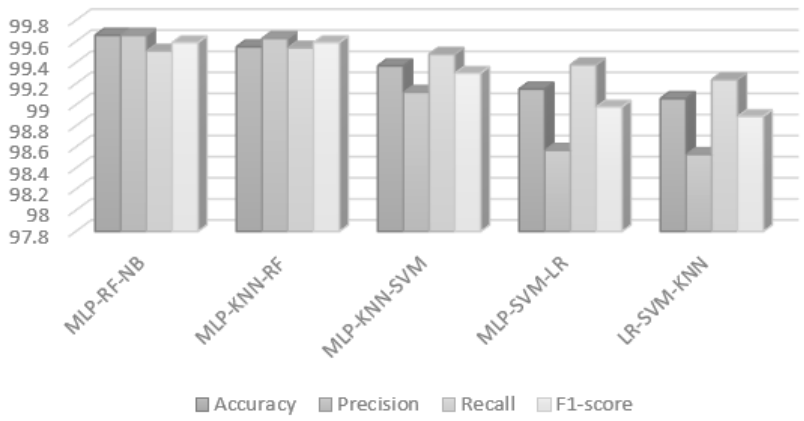


Figure 15. Comparison of the five proposed ensemble learning models using the most known indicators (%)

To prove that our approach can achieve robust performance and provide many competitive strong points when compared to other existing approaches. The figure below depicts a comparison of the proposed model that has reached the highest accuracy (MLP-RF-NB) and other recent research studies that have also been based on the ensemble technique using the same data set as we have opted for.

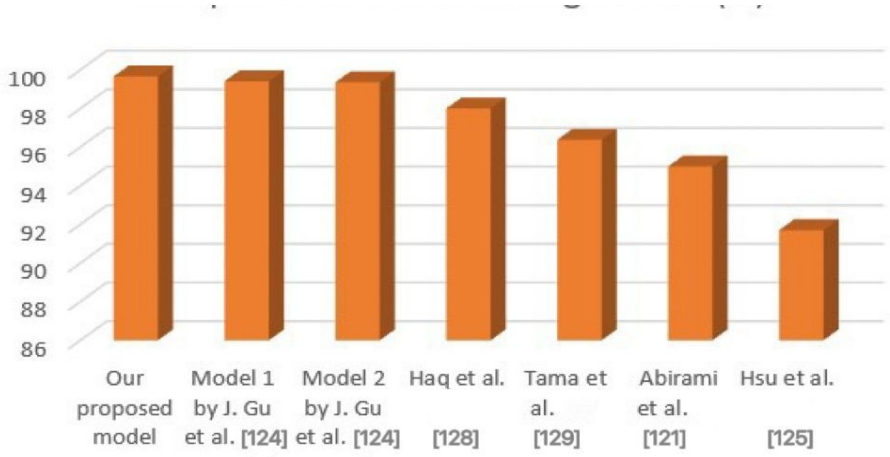


Figure 16. Accuracy rate of the proposed method and other existing approaches

Using Tesla T4 in Google Colab has not helped a lot in enhancing the prediction speed, it is still not an appropriate solution to make the models fast and efficient. Figure 17 shows that the combination of LR, SVM and, KNN takes less time for prediction, compared to all other models as it takes 66.06 seconds. This issue will be improved in the upcoming work.

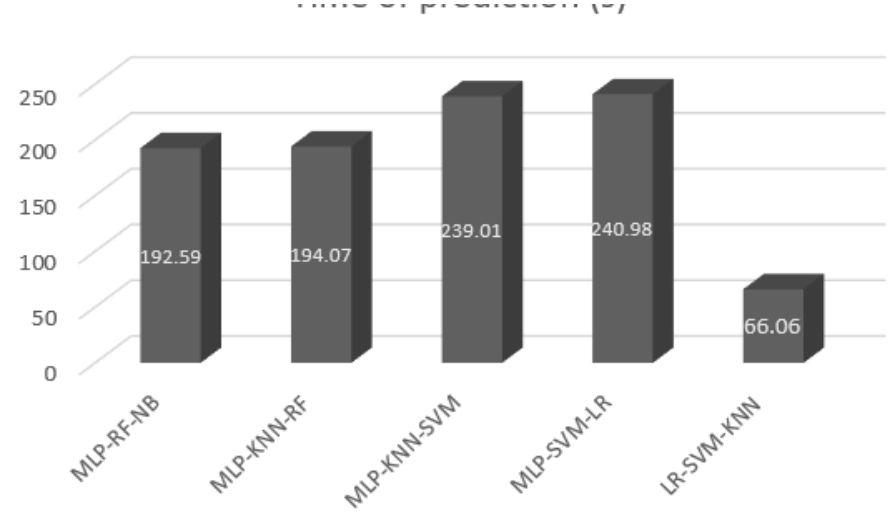


Figure 17. Prediction time of the 5 proposed ensemble learning models in Seconds

3.6 Conclusion and Future Scope

Various intelligence algorithms are taking the intrusion detection task in new different dimensions, aiming to develop and update the detection competence of the intrusion detection

systems that play a vital role in the cybersecurity domain. Among these algorithms, the ensemble learning approach has received special attention from many researchers who approved its great capacity in increasing the prediction accuracy and reducing false positive rates. In this paper, we propose 5 network intrusion detection systems based on a stacked ensemble technique that combines the strong points of different machine learning classifiers: Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), K-nearest neighbors (KNN), Naïve Bayes (NB) and Multilayer Perception (MLP), in order to produce a better prediction and deal with the issues that an IDS encounters each year due to the current security challenges. The models were evaluated through the NSL-KDD data set and compared them to each other to come out with the best combination in terms of accuracy, precision, recall, F1-score, and time of prediction. The experimental results show that the ensemble model of MLP, RF, and NB outperforms all other combinations and other similar existing research studies, as it achieved an accuracy of 99.66%. Furthermore, the quality of the methodology used in this work also helped to provide a high classification accuracy with the limitation of false positive and false negative predictions. In our future work, we plan to expand our experimentation to classify several types of cyberattacks in real-time.

Chapter IV

Importance of Hybridization Modeling for Strengthening the Network Intrusion Detection

4.1 Introduction

The security of computer networks is crucial in today's interconnected digital environment. The need for efficient Intrusion Detection Systems (IDS) has substantially increased due to the complexity and sophistication of cyber threats. Traditional rule-based and signature-based IDS solutions frequently have trouble keeping up with attack methods that are continually developing **[140]**. Therefore, there is a strong case for using more sophisticated and adaptable algorithms that can efficiently identify both known and undiscovered assaults.

Hybridization learning, which combines the benefits of many machine learning methods to provide a more reliable and accurate detection system, is one promising strategy to improve the performance of an IDS **[141]**. AdaBoost (Adaptive Boosting) stands out among the numerous ensemble learning techniques as an effective and popular strategy for building a potent hybrid model **[142]**. It is a common ensemble learning strategy that excels at building a robust classifier by repeatedly highlighting the cases that were incorrectly classified and instructing succeeding classifiers to avoid making the same mistakes. AdaBoost is very well suited for hybridization learning in network intrusion detection systems because of its versatility **[143]**.

Based on previous research works, we have been convinced enough to rely on AdaBoost algorithm in combination with MLP, RF, and SVM, in order to approve the efficiency of ensemble learning in maximizing the performance of a NIDS. Moreover, an analytical study is also provided to discuss the difference between using single and ensemble learning techniques.

The experiments are evaluated within the NSL-KDD data set using the feature selection technique to boost the quality of the prediction process. Furthermore, this work reveals the weaknesses of each method used, which will lead to a new approach for the next work that can deal with these limitations.

4.2 Related works

Much effort has been devoted to improve the intrusion detection system sector. Many researchers have integrated machine learning algorithms to make the IDSs more performing and appropriate for the current security challenges. In this section, we have classified these efforts into two well-known learning methodologies adopted by different research studies:

4.2.1 Single learning method

As far as is known, among the most efficient and widely used machine learning algorithms are Artificial Neural Networks **[144]**. Peng et al. approved the importance of the latter by proposing an approach based on MLP, LSTM and other ANN algorithms **[145]**. The purpose of their contribution is building an intelligent model, which can identify the SQL injection attack within the HTTP traffic data in real time. Their obtained results show how a MLP classifier with 3 hidden layers performs well in achieving the main aim of their work. Additionally, their new approach is able to detect the unknown attacks and can also handle the incomplete blacklist filtering issue.

In another work, there is a new architecture of ANN based on IDSs, put forward by Shenfield et al. **[146]**, which aims to identify the malicious activity and distinguish between a harmful and a benign threat within complex network traffic. The authors came out with a

satisfying result in terms of classification, accuracy, precision and sensitivity over 1000 iterations of repeated 10-fold cross-validation, which proves the robustness of their detection system.

Naïve bayes is also a popular classification algorithm, which is known by its simplicity **[147]**. For instance, Panda et al. have developed this classifier and have been based on it to build a robust IDS model. They have proposed a 2-level classification approaches. The first level presents the base classifier using Discriminative Multinomial Naive Bayes (DMNB). In the second level, they used the nominal-to-binary supervised filtering with 10-fold cross validation for experimental results over NSL-KDD dataset, which reveals that the proposed approach works better than other ML algorithms such as Decision Tree, Random Forest, and Support Vector Machine.

In addition, Sravani et al. **[148]** confirmed and proved that Naïve Bayes and Neural Network algorithms are effective to classify and detect intrusions, with a focus on the abnormal ones. Their results proved that both of them decreased the false negatives and false positives. Besides, they increased the accuracy of their developed IDS.

Random Forest is also an important classification algorithm, which has been enhanced in **[134]**. The authors attempted to detect four different types of attacks using NSL-KDD dataset and evolved the accuracy of their proposed model that was good in decreasing low alarms and increasing detection rate.

Furthermore, many analytical and comparative studies have been carried out. They discuss the advantages and the limitations of each classifier, which will result in the selection of the most performing algorithm. For example, Belavagi et al. built an enhanced IDS model using different steps **[137]**. They have been based on four ML classifiers namely, Random Forest, Logistic Regression, Support Vector Machine, and Gaussian naïve Bayes. Their results proved that RF is

better than other classifiers in terms of Accuracy, Precision, Recall, and F1-score. Besides, LR reached a better accuracy than SVM and GNB.

In another comparative study, Pal et al. used two algorithms: RF and SVM, to improve the performance of an IDS. Their outcome showed that SVM reached the highest accuracy rate in detecting anomaly traffic within the KDD'99 dataset. However, RF takes less time to make predictions and it has better precision compared to SVM **[149]**.

As can be deduced, each machine learning algorithm has its weaknesses and strengths. There are several types of cyberattacks, including SQL injection, Phishing, Denial of service, man in the middle, eavesdropping attack, etc., Hence, some classification algorithms can be performant in detecting a type of attack, but cannot perform well on another type **[150]**. As a result, the single learning method cannot always be considered as a sufficient solution to provide a strong security. For this reason, numerous research studies opt for collaborative learning.

4.2.2 Collaborative learning method

Many researchers approved that the collaborative learning method can achieve high accuracy rate compared to the single learning technique. Because it deals with the statistical and computational problems, by decreasing variance and bias **[151]**. Consequently, it provides better predictive results. Khonde et al. validated the efficiency of the ensemble method and applied it to boost the accuracy of an IDS in detecting normal and malicious activities **[152]**. They combined 5 classification algorithms: SVM, DT, RF, KNN and ANN, using a novel method of feature selection. The latter had a significant capability in examining real time traffic. Therefore, their method performs well and produced good results.

Das et al. have also relied on the collaborative approach and grouped 4 different algorithms, namely; J48, SVM, MLP, and KNN in order to build a powerful IDS **[153]**. Their

proposed method unfolds into two stages: Firstly, during the training step each algorithm create a model, which is able to detect DDoS attack and its recent unknown types. Secondly, the final model which combines the trained algorithms is evaluated in the test set using 10-folds cross validation and feature selection techniques. Their experimental results showed the important potential of their model in detecting new types of DDoS attack. Besides, it outperforms many other existing models.

In another work, an enhanced IDS model has been created in **[154]** using the fusion of the hybrid feature selection technique and 2-level classifier ensemble. The latter is relying on 2 meta learners (rotation forest and bagging) that are evaluated in two different datasets: UNSW-NB15 and NSL-KDD. The obtained results of their proposed model validated the importance of the ensemble learning in maximizing the accuracy rate. In addition, the authors have considered the training time as a significant measurement that can impact the performance of the model. Thus, it should also be taken into consideration in order to improve the intrusion detection task.

To strengthen the ability of an IDS in providing a strong security, Lian et al. have been based on the stacking/blending technique, which is a 2-levels ensemble learning method. This necessitates training different single learning algorithms on the training data. Then, it combines the outputs of each algorithm for the final prediction task over an unseen data **[155]**. The authors used the decision tree-recursive feature elimination (DT-RFE) algorithm, which helps reducing the system size and the time required for training **[156].** Therefore, their designed IDS achieved some excellent results in terms of detection rate accuracy for the classification of 4 types of attack (Dos, Probe, U2R and R2L). Moreover, it has low false positive rate.

In **[157],** a new deep learning approach has been proposed using AdaBoost Regression algorithm to improve the quality of cyberattacks detection task and ensure a strong network

security in real time. The authors approved based on their obtained results that AdaBoost performed effectively for the main aim of their contribution. This classifier boosts the classification accuracy by identifying the weak learners based on the decision tree concept with the consideration of different levels.

4.3 Proposed Hybridization Modeling

It is well-known that real-world data usually contains redundant values, outliers, etc., which results in the complexity of analyzing the data and the degradation of the accuracy rate. As a matter of fact, data preprocessing is considered as an important process. In this contribution, we have followed the process illustrated in the flow below, knowing that we used the same technical tools and dataset of the previous ensemble learning study.

→ **Data-preprocessing:** This phase plays a vital role in removing redundant entries. As is shown in Figure **18**., it includes the distribution of each attack type to its attack class, which represents the target variable. Besides, to remove the mean and unit variance, a scaling of numerical attributes is applied. Then, the categorical attributes are encoded for data sampling and resampling techniques. Thus, the dataset will be more balanced.

→ **Feature selection:** In order to reduce the computational cost of modeling and enhance the performance of the classification process, in this phase, we have minimized the features of the dataset and kept 10 significant variables, that have a direct impact on the target variable *(attack class)*.

→ **Classification:** Based on some analysis studies of several machine learning algorithms **[158, 159]**, which considers the same dataset, it was noticed that Random Forest and Multilayer perceptron reached the highest accuracy. For this reason, this paper considered these two classifiers in conjunction with the Adaboost algorithm **[160]**. The latter has a

vital role in maximizing the performance of the IDS model. It boosts the weaknesses of other algorithms, which results in the creation of a powerful hybrid model.

→ **Evaluation:** The performance of the hybrid model has been examined using the accuracy, precision, recall, f1-score, and time of prediction for binary classification.

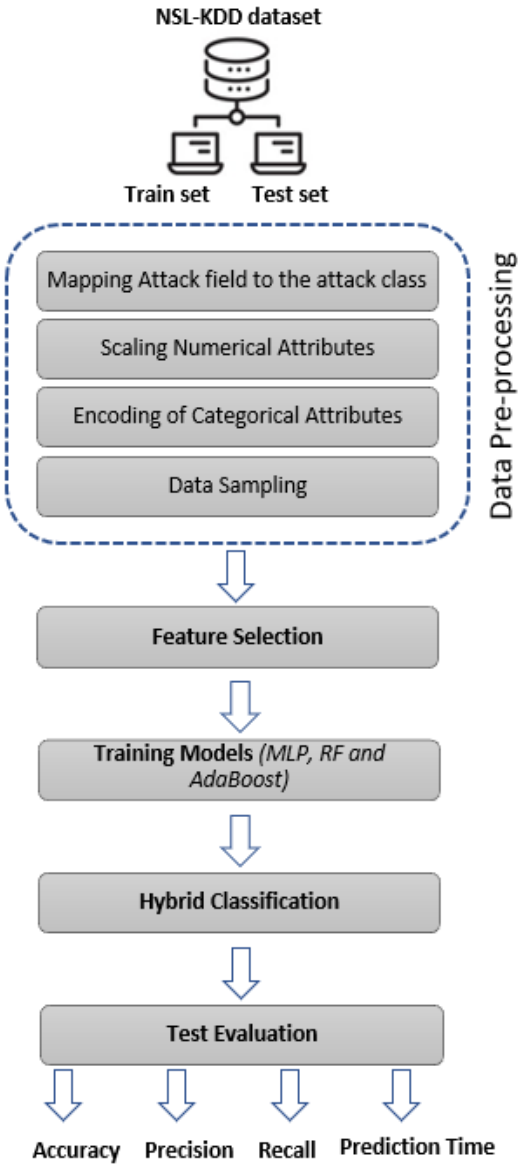


Figure 18. Flow chart representing the proposed methodology

4.4 Results and Discussion

In this proposed study, we approve the effectiveness of the hybridization method based on the collaboration of Multilayer Perceptron and Random Forest with Adaboost classifier. From the results obtained, it can be validated that this technique is positive for boosting the performance of an IDS model based on machine learning algorithms in detecting normal and abnormal activities within the network. As is indicated in the table below, our hybrid model outperforms other models based on single classifiers. It has achieved the highest accuracy rate of 99.76% in detecting a wide variety of attacks *(Dos, Probe, R2L, and U2R),* categorized in attack class and also normal behaviors. Moreover, it returns very low false positives, since it has a precision of 99.64%. Besides, it obtains a percentage of 99.80% in terms of false negative identification. Considering the same methodology and dataset, the MLP and RF have reached an accuracy rate of 91.41% and 82.99%, respectively. Moreover, they are generating a great portion of false positives and have a weak identification of true positives.

Table 6. The performance of the proposed hybrid IDS model compared to single classifiers over the test set (%)

Classifiers	*Accuracy*	*Precision*	*Recall*	*F1-Score*
MLP	91.41	80.71	96.68	87.98
RF	82.99	76.71	82.85	79.67
Our hybrid model	**99.76**	**99.64**	**99.80**	**99.72**

Further, from the results provided in the figure below, which presents the time of prediction required by each model for the detection task, it is evident that despite the highest accuracy reached by our model, it takes a lot of time to make predictions. It is a computationally costly process. However, RF is the less time-consuming classifier in this work, as it takes 4.59e-05

Milliseconds and MLP reached the second rank since it takes 1.00e-4 Milliseconds for the prediction process.

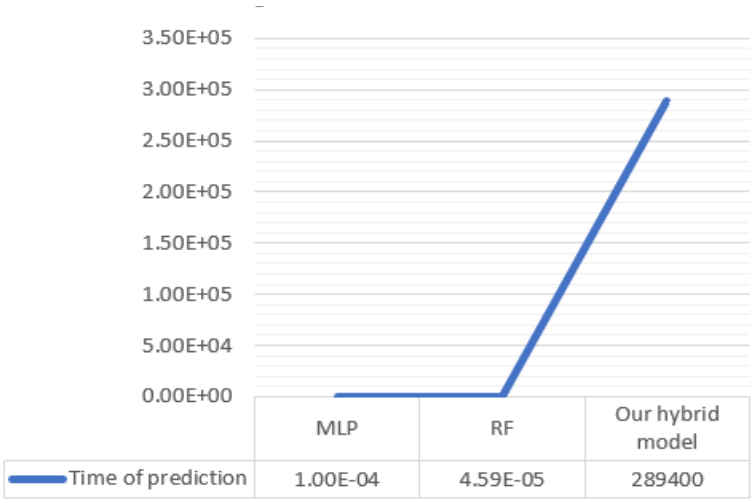


Figure 19. Time required by each algorithm for the prediction process in Ms

To prove the effectiveness of our hybrid model, we compared it to some existing hybrid models, as Figure 20 depicts.

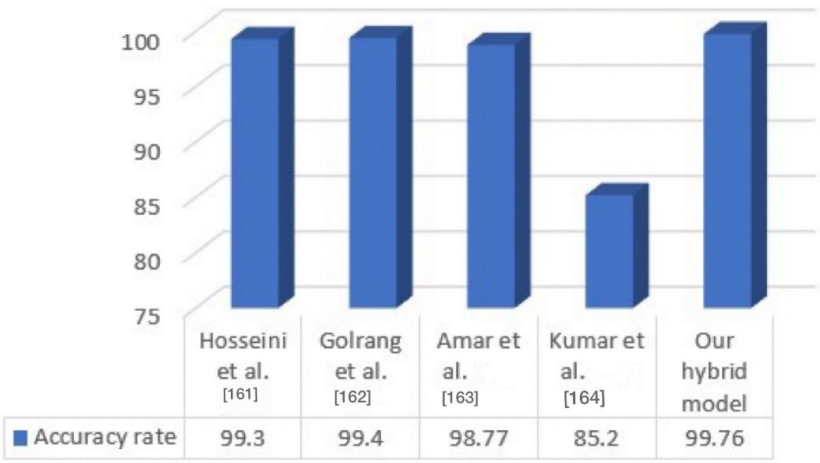


Figure 20. Comparison of the proposed hybrid model to another existing model (%)

It is evident how this combination of Multilayer Perceptron, Random Forest, and Adaboost algorithms comes up with a high-performance intelligent hybrid model for developing the task of

intrusion detection within computer networks. It provides a better accuracy rate with less false alarms, compared to other approaches, as the detection accuracy in **[161], [162], [163],** and **[164]** is reaching 99.3%, 99.4%, 98.77%, and 95.2%, respectively. Whereas our approach achieved 99.76%. However, the limitation of the proposed hybridization technique is the time required for the predictions.

4.5 Conclusion

An enhanced machine learning-based modeling has been proposed in this paper, in order to strengthen the performance of a Network Intrusion Detection System (NIDS) for a binary class classification (i.e., abnormal and normal behaviors). It is a hybridization approach based on Adaboost algorithm, which combines two other different machine learning classifiers, namely: Multilayer Perceptron and Random Forest, using the feature selection technique. The performance of the latter was assessed on the networking NSL-KDD dataset and the achieved outcomes approve the importance and efficiency of the designed hybrid model since it has maximized an accuracy of a single model from 91.41% and 82.99% to 99.76%. This paper also takes into account an analytical study, which compares the hybridization and the single learning methods, considering the same methodology and dataset. Thus, a more accurate picture of the limitations of each method and algorithm used is provided. This will be helpful to propose novel techniques in order to handle these weaknesses, which include the time required by the model to be built and to make predictions. In addition, the presented model has been compared to other existing hybrid methodologies for network IDS and it is shown that the proposed hybridization method is more efficient. As future work, the limitations of the presented cybersecurity modeling will be overcome. It will also be implemented for other internet technologies and applications.

Chapter V

i-2NIDS Novel Intelligent Intrusion Detection Approach for a Strong Network Security, Dealing with Imbalanced Classification Issue

5.1 Introduction

An IDS model cannot make good predictions when the parameters are incorrectly selected and also because of the classification issues, such as; underfitting, overfitting, imbalanced data, etc. For this reason, there is a useful technique, namely; cross-validation. It is considered as a resampling procedure for the determination and the selection of the appropriate parameters, which cost the least test error. It is a well-known evaluation method for machine learning models that shows how well the latter will perform to an independent test data that has not been used during the training phase of the model **[165]**. This approach proceeds by splitting the cleaned dataset into k-chunks of equal size. The first partition is considered as a validation set, and the model is fitted on the remaining k-1 partitions that present the training partitions. Then, the analysis is performed on each fold. Finally, it takes the average of scores of all partitions, which presents the overall estimate error. Hence, the cross-validation technique provides a better utilization of the data and it comes in different types. The most commonly used are presented in the following table.

Table 7. Types of cross-validation technique

Cross-validation types	Description	(+)	(-)
Holdout cross-validation	The simplest type of cross-validation approach. It randomly separates the data into training and test sets. The more data is used for the model's training, the better its performance will be.	Simple to implement	Doesn't work well for imbalanced data
K-Fold cross-validation	The dataset is equally split into k folds, then the holdout approach is repeated k-times until each fold is considered as a test set and other k-1 folds as training set.	Less time consuming	Doesn't work well for imbalanced data
Stratified K-Fold cross-validation	The dataset is divided into k partitions so that the validation set has an equal instance of the dependent class label, which is a good solution for an imbalanced dataset. Then, it computes the final score based on the mean of scores of each partition.	Help to reduce Bias and Variance	Suitable for imbalanced data
Leave-P-Out cross-validation	It considers p observations as a validation set and p-1 data as a training set. This process is repeated for all p combinations. Then, it averages the accuracies from all iterations to deduce the final accuracy.	The bias will be lower	Computationally complicated
Leave one out cross-validation	A less exhaustive method, because it is considered as a simple variation of the previous cross-validation type, as the value of p is set as 1.	Simple to implement	Low bias issue
Repeated random sub-sampling validation	Also called Monte Carlo method; it divides the dataset randomly into k-folds for training and validation. K is number of times the model has been trained. The final score average is obtained as the mean from the number of repeats.	Training and validation splits are independ -ent of the iterations number	Doesn't work well for imbalanced data
Nested cross-validation	It is considered as a technique to tune the parameters of an algorithm, unlike the other cross-validation methods that only aim to estimate the performance of an algorithm.	Reduce the bias	Computationally expensive

There are several security concerns and classification problems that directly affect the performance of an IDS. Our contribution to the cybersecurity field is dealing with these issues and strengthening the ability of an intelligent IDS in detecting anomalies and various types of cyber-attacks within the network environment. The proposed methodology of this paper is based on the nested cross-validation technique, which can properly tune the parameters of a machine learning algorithm and concurrently generalizes the error estimate. It was evaluated over the NSL-KDD dataset since the latter eliminates repeated and redundant records. Hence, the issue of biasing the features and classifiers has been resolved by this dataset **[166]**. The presented results show how this proposed approach has been well-performed and efficient to achieve the main purpose of this research study, in terms of accuracy, precision, recall, and f1-score.

5.2 Related Works

Based on a proposed approach of Amit et al. about recognition with randomized trees in 1997 **[167]**, Breiman came out with an ensemble method, which is able to make predictions relying on the majority vote. The latter is the result of multiple collections of decision trees that have voted for the most popular class. Each tree is dependent of the values of an independent sampled random vector, with similar distribution for the combination of trees constituting the forest **[168, 169]**. Random Forest can be used for the classification and also the regression. Moreover, its default hyperparameters frequently result in high accuracy prediction rate. Several researchers approved this through different comparative studies. For instance, Fedaku et al. conducted an analytical performance evaluation of this algorithm and compared it to 4 well-known classifiers, such as Logistic Regression, Support Vector Machine Stochastic Gradient Decent, and Sequential Model over the NSL-KDD dataset **[123]**. The obtained results validated the efficiency of Random Forest against these algorithms and showed its ability to deal with

noise and over-fitting issues in data. An enhanced version of Random Forest has been proposed by Farnaz et al., confirmed the great potential of this classifier in detecting 4 types of attack (DoS/DDoS, Probe, R2L, and U2R) with very low false alarms. Their experimental results show that RF performs well than other traditional machine learning algorithms **[134]**. Belavagi et al. have also approved in their comparative study the flexibility and the powerful performance of RF compared to Logistic Regression, Support Vector Machine, and Gaussian Naïve Bayes **[137]**.

While LR is a predictive analysis algorithm, which is also called Logit Regression relating to the logistic function used at the core of the technique. This classification algorithm is based on the probability estimation concept and used to allocate the observations to a specific class. It works by transforming the output using the logistic function in order to map predicted values to probabilities **[170]**. There are two types of classification that can be made by Logistic Regression: Binary classification, which requires classifying between only two cases (e.g., normal or abnormal activity) and there is the multiclass classification, which is about classifying between more than two cases (e.g., Dos/DDoS, Probe, R2L, U2R or normal activity). Generally, researchers apply this algorithm due to its fast classification for the unknown records. Additionally, it is known to be simple and easy to implement and train. Faizal et al. have suggested a new approach that can identify a fast attack intrusion based on the number of connections made in one second **[171]**, the results were evaluated using LR classifier and validated that it is an appropriate method to determine a network traffic whether it is a malicious activity or not. Belavagi et al. have also been based on LR algorithm in order to build an enhanced IDS model and it has reached a better accuracy compared to Support Vector Machine and Gaussian Naïve Bayes **[137]**.

Another popular type of Artificial Neural Networks models (ANN) namely, MLP, used for recognition, approximation, classification, and prediction tasks. It is composed of interconnected nodes that are called neurons in three types of layers: input layer, hidden layer, and output layer **[172]**. The input layer presents the input data to process. A random number of hidden layers are located between the input and output layers, the true values of their nodes are unknown in the training dataset, for this reason, they are called "hidden". The requested task accomplished by the MLP algorithm is performed in the output layer. Each neuron is connected to the preceding neuron as well as the following one and the connections between them are indicated by a weight. The latter is computed based on a learning algorithm. Multilayer Perceptron has the ability to handle large amount of data. Furthermore, it works well in solving complex non-linear issues **[173]**. Therefore, Aurora et al. have been based on MLP neural network and other algorithms: Logistic Regression (LR), Voted Perceptron (VPP), and Radial Base Function (RBF) over the NSL-KDD dataset, in order to improve the intrusion detection task **[174]**. The obtained results of their contribution showed that MLP outperforms all other algorithms in terms of accuracy. In **[175]**, the authors proposed an approach based on different ANN models, such as MLP and Long Short-Term Memory (LSTM), aiming to identify in real-time the SQL injection attack on HTTP traffic data **[175]**. The experimental results approved that MLP using 3 hidden layers has been the best in detecting unknown attacks and handling the incomplete blacklist filtering issue.

Furthermore, a distance-based algorithm called KNN, can be used to deal with both regression and classification problem statements. It calculates the distance from the nearest neighbors located in the proximity of the unseen data and extracts those with the shortest distances to it, in order to determine what class this unknown variable belongs to. 'K' is the hyperparameter for this approach and it is referred to the number of nearest neighbors to an

unseen variable that has to be predicted. For the regression issue, KNN relies on the mean/average technique. It calculates the mean for the values of the closest neighbors until all the latter have been identified in a certain range of the value 'K'. Otherwise, if the problem statement is a classification task, KNN employs the majority vote concept. Based on the given range of 'K' values, the class that has achieved the most votes, is selected **[176]**. It is a simple algorithm to interpret and it also does not require much time to build the model, Kumar et al. have approved these advantages of KNN in their contribution **[177]**. The authors have developed the performance of an IDS using KNN and SVM algorithms within the KDD'99 dataset to select the best classifier. The obtained results show that KNN has reached a higher accuracy and performs better when it is employed with the principal component analysis (PCA). Another proposed approach by Wazirali about improving an IDS has been based on the KNN hyperparameter tuning with 5-fold cross-validation **[178]**. The author carried out a comparative experiment on the NSL-KDD dataset with different existing methods and it is shown that it is an effective approach since it has the highest detection rate.

5.3 The Proposed i-2NIDS and Methodology

The proposed approach is a hyperparameter optimization methodology, based on a feature selection technique to enhance the accuracy of an IDS using machine learning classifiers. It is generally known that these latter regularly encounter the overfitting issue, which negatively impacts the performance of the models, since it makes them fit more data than they actually necessitate. As a consequence, they capture meaningless values and noisy data in the training set. Hence, the model becomes overfitted and cannot generalize unseen data **[179]**. To avoid this problem and increase the accuracy and efficiency of the security models, the suggested technique is opting for the nested cross-validation method.

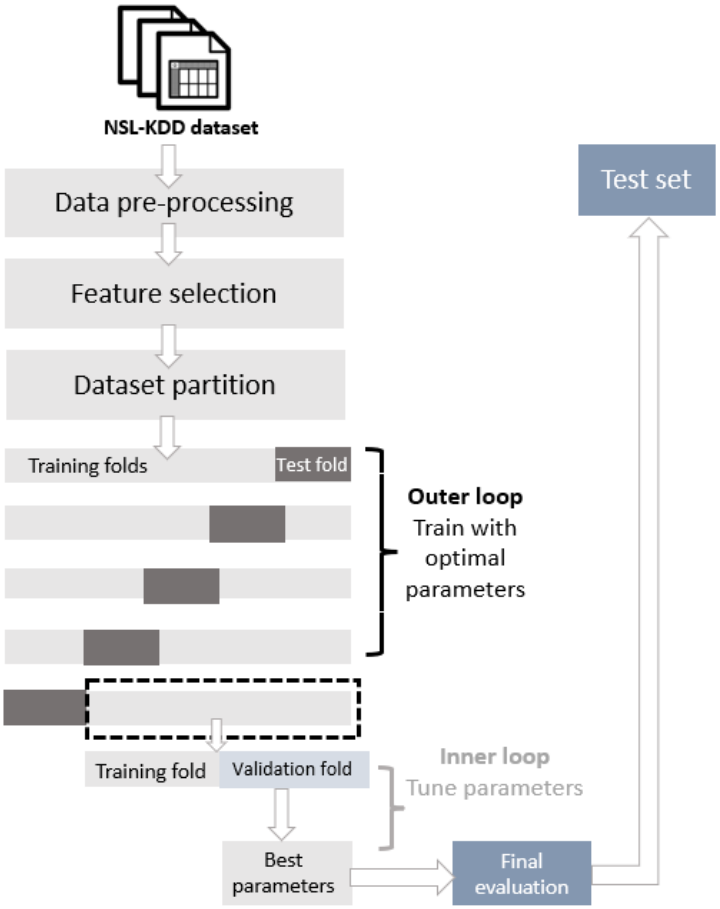


Figure 21. The architecture of the proposed model

Data Preprocessing

As is shown in Figure 2**1.** At first, the pre-processing method is applied to the whole dataset. Initially, it consists of mapping the attack field to the fourth attack class *(Dos, Probe, R2L, and U2R)*, that represent the target field. It helps organize numerous bits of data into a system that is capable of being handled and managed. Then, the redundant and duplicate fields have been removed to avoid biasing the fitted model and reducing the negative impact of overfitting. Afterward, the numerical features are extracted to apply on them the scaling method,

which is also known as data normalization, aiming to eliminate the mean and unit variance. Finally, to address the problem of imbalanced classification in both attack classes representing R2L and U2R; as it is shown in Table **8**., it is important to randomly resample the training dataset. In this case, we have applied the oversampling method, which necessitates duplicating examples from the minority class in the training dataset. This was performed on the categorical data that has been encoded beforehand.

Table 8. Attack class distribution in the NSL-KDD dataset

	Attack class	**Frequency train (%)**	**Attack class**	**Frequency test (%)**
DoS/DDoS	45927	36.46	7458	33.08
Probe	11656	9.25	2421	10.74
R2L	995	0.79	2754	12.22
U2R	52	0.04	200	0.89
Normal	67343	53.46	9711	43.08

Feature Selection

After preparing the data and making it clean and suitable for the proposed model, the following step is necessary to perform the feature selection technique. The purpose of this aspect is to select the most significant subset of the input variables that enhances the ability of a learner algorithm to efficiently classify patterns. It is also called the dimensionality reduction method since it minimizes the large number of attributes. Hence, it reduces the computational cost of modeling and maximizes the accuracy of the classification rate. In this work, the features of the dataset have been reduced from 42 to 10 attributes that are having a direct impact on the dependent variable.

Table 9. Top 10 most important features selected in the NSL-KDD dataset

Attribute Name	Description
'src_bytes'	The sent data bytes in single connection from the source to the destination.
'dst_bytes'	The sent data bytes in single connection from the destination to the source.
'logged_in'	It indicates the login status (1 or 0).
'count'	Connections number to the same destination host as the current connection in the 2 previous Seconds.
'srv_count'	Connections number to the same service as the current connection in the 2 previous Seconds.
'dst_host_srv_count'	Connections that have the same port number.
'dst_host_diff_srv_rate'	Connections that were to the same service among the connections in *'dst_host_count'*.
'dst_host_same_src_port_rate'	Connections that were to the same source port among the connections in *'dst_host_srv_count'*.
'dst_host_serror_rate'	Connections that have activated the flag s0/s1/s2 /s3 among the connections in *'dst_host_count'*.
'service'	The used destination network service.

Data pre-processing for training dataset based on nested cross-validation

Firstly, each algorithm has to include the hyperparameters for tuning. The main idea of using nested cross-validation is adding another loop to the cross-validation procedure. The outer loop is defined as a train/test split proceeding, whereas the inner loop is about selecting the best hyperparameters setting from the training folds, that take place in the outer loop. The latter in this suggested method is having a 5-fold cross-validation and in each fold the dataset is split into a combined training folds and a test fold, as illustrated in Figure **21**. It presents the same procedure as the k-fold cross-validation method. Expect that in the latter, the model is fitted on the training folds and evaluated on the test fold. In this case, another K-fold cross-validation

method is considered on the fifth training folds that are split into two partitions: The training fold and validation fold, constituting the inner loop to apply the hyperparameter tuning with the aim of finding the best settings. Finally, we can evaluate the models within the test set.

Performance Evaluation

Also in this paper, we used the same performance measures as the previous contributions presented in Chapters **3** and **4** (Accuracy, Precision, Recall, and F1-Score). In this work, we have evaluated the results using both binary *(normal or abnormal)* and a multi-label classification confusion matrix of 5 classes: Normal, Dos, Probe, R2L, and U2R, as shown in the table below.

Table 10. Confusion matrix for 5-class classification

		PREDICTED CLASS				
		NORMAL	***DOS***	***PROBE***	***R2L***	***U2R***
ACTUAL CLASS	*NORMAL*	C1	C2	C3	C4	C5
	DOS	C6	C7	C8	C9	C10
	PROBE	C11	C12	C13	C14	C15
	R2L	C16	C17	C18	C19	C20
	U2R	C21	C22	C23	C24	C25

In the case of multiclass classification, considering the example of Normal class, the TP, FP, FN, TN are calculated as follows:

TP: The predicted value and actual value should match.

$$\mathbf{TP = C1}$$

FP: The total of all values of the corresponding column, except the TP value.

FP = C6+ C11+C16+C21

FN: The total of all values of corresponding rows, except the TP value.

FN = C2+C3+C4+C5

TN: The total of all values composing the confusion matrix, expect those of all rows and columns corresponding to the class in question.

TN = C7+C8+C9+C10+C12+C13+C14+C15+ C17+C18+C19+C20+C22+C23+C24+C25

Similarly for the remaining classes.

5.4 Experiments and Discussion of Results

In this section, firstly, the technologies and tools used in the experimental phase are briefly described. Secondly, an analytical comparison of the evaluation of the 4th classifiers used for the proposed approach over multi-class classification strategy and binary classification is discussed in terms of accuracy, precision, recall, and f1-score. Finally, the proposed model, based on the best-selected classifier by the nested-cross-validation method, is compared to recent intelligent IDS models, considering the same experimental dataset.

Tools and Environment

This research was performed using a Dell Inspiron 15 3000 Series Core i5 7th Gen., and experimented on Anaconda, which is a Python distribution and open-source software that comes with more than 1400 packages needed for artificial intelligence and data science experimentations. This platform allowed us to build our proposed intelligent IDS on the Jupyter Notebook, which is a flexible JSON document composed of Input/Output cells of a Python

script. It is known to provide performing computations, plots, and other media types within the browser of the user's system **[180]**.

Experimentation Results

Multiclass classification

According to the obtained results in Tables **11**, **12**, **13**, **14**, and **15**, we can notice that Random Forest outperforms all other algorithms in terms of all the performance indicators within the test dataset. It has reached an accuracy of **99.97%**, precision, recall, and f1-score of **99.96%** to detect normal activities. Such challenging outcomings are due to the optimal selection of the hyperparameter for the classification model. The best parameter considered for RF in this study is *'n_estimators'* set to '500', which presents the number of trees built before taking the averages of predictions. These results approve the robustness of the model. Additionally, the latter yield the highest accuracy of **99.79%** to detect DoS/DDoS attacks with a precision of **99.79%**, a recall of **96%,** and f1-measure of **99.88%**. Moreover, when it comes to detecting the Probing attack, RF performs better than other classifiers, since it has an accuracy of **99.94%**, precision of **99.92%**, recall of **94.48%,** and f1-score of **99.73%**. Despite the imbalanced class distribution issue encountered by R2L and U2R classes, which are intrinsically rare, our proposed model based on RF has overcome this challenge and achieved the best accuracy of **99.96%** to detect R2L attack class with no false positives, as it has a precision of **100%**. Besides, it has the highest recall rate of **94.48%** and f1-score of **97.17%**. Furthermore, the RF and KNN are considered to be similar in detecting U2R attacks. They both achieved an accuracy of **99.98%**, precision of **71.43%**, recall of **50%** and f1-score of **58.83%**.

The MLP has ranked second with an accuracy of **99.92%** and a precision of **99.90%** in

detecting normal class. The optimal parameters selected for this algorithm are the activation function for the hidden layer set to *'tanh'* (hyperbolic 'tan' function), the solver for weight optimization set to *'adam'* (a stochastic gradient-based optimizer), the hidden layer sizes set to *'100'*, the learning rate schedule for weight updates is set to *'adaptive'* (as long as the training loss is decreasing, this parameter keeps the learning rate constant to the initial learning rate) and the regularization term that handle the overfitting issue is set to *'0.0001'*. However, in most of cases, the KNN performs better than MLP in detecting DoS/DDoS, Probe, R2L, and U2R. The best parameters considered for the KNN are the number of neighbors to use by default for k-neighbors queries, which is set to *'1'*, and the power parameter for the Minkowski metric set to *'1'* (equivalent to the application of Manhattan distance).

The classifier with the low performance in predicting normal class is LR with an accuracy of **98.38%**, precision of **98.39%**, recall of **97.13%,** and f1-score of **97.78%**, based on setting the penalty parameter to 'l2' (default choice) and the tradeoff parameter 'C' of the algorithm that determines the strength of regularization is fixed to '1000.0'. However, the precision of this classifier in predicting the U2R attacks is **55.56%**, which is higher than the precision of MLP as it has only reached **35.30%**.

Table 11. The performance metrics of the proposed models for a multi-class classification detecting Normal class (Test results %)

Algorithms	Accuracy	Precision	Recall	F1-Score
RF	99.97	99.96	99.96	99.96
KNN	99.91	99.81	99.93	99.87
MLP	99.92	99.90	99.89	99.90
LR	98.38	98.39	97.13	97.78

Table 12. The performance metrics of the proposed models for a multiclass classification detecting DoS/DDoS Attack (Test results %)

Algorithms	Accuracy	Precision	Recall	F1-Score
RF	99.79	99.79	99.96	99.88
KNN	99.64	99.62	99.72	99.67
MLP	99.50	99.57	99.49	99.53
LR	95.30	95.05	96.23	95.64

Table 13. The performance metrics of the proposed models for a multiclass classification detecting probe Attack (Test results %)

Algorithms	Accuracy	Precision	Recall	F1-Score
RF	99.72	99.92	99.53	99.73
KNN	99.82	99.14	98.85	98.99
MLP	99.77	98.76	98.76	98.76
LR	97.18	85.93	83.06	84.48

Table 14. The performance metrics of the proposed models for a multiclass classification detecting r2l Attack (Test results %)

Algorithms	Accuracy	Precision	Recall	F1-Score
RF	99.96	100	94.48	97.17
KNN	99.89	95.68	88.95	92.20
MLP	99.84	98.90	85.43	89.01
LR	99.45	62.89	72.37	67.30

Table 15. The performance metrics of the proposed models for a multiclass classification detecting U2R Attack (Test results %)

Algorithms	Accuracy	Precision	Recall	F1-Score
RF	99.98	71.43	50	58.83
KNN	99.98	71.43	50	58.83
MLP	20.12	35.30	60	44.45
LR	99.97	55.56	50	52.64

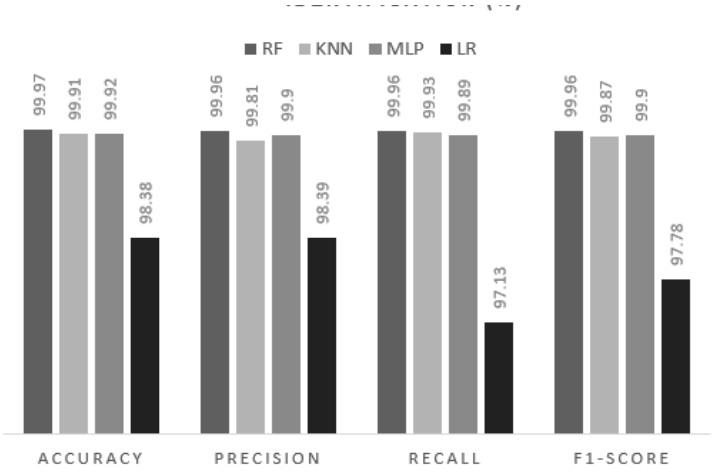


Figure 22. The performance metrics of the proposed models for normal class identification (%)

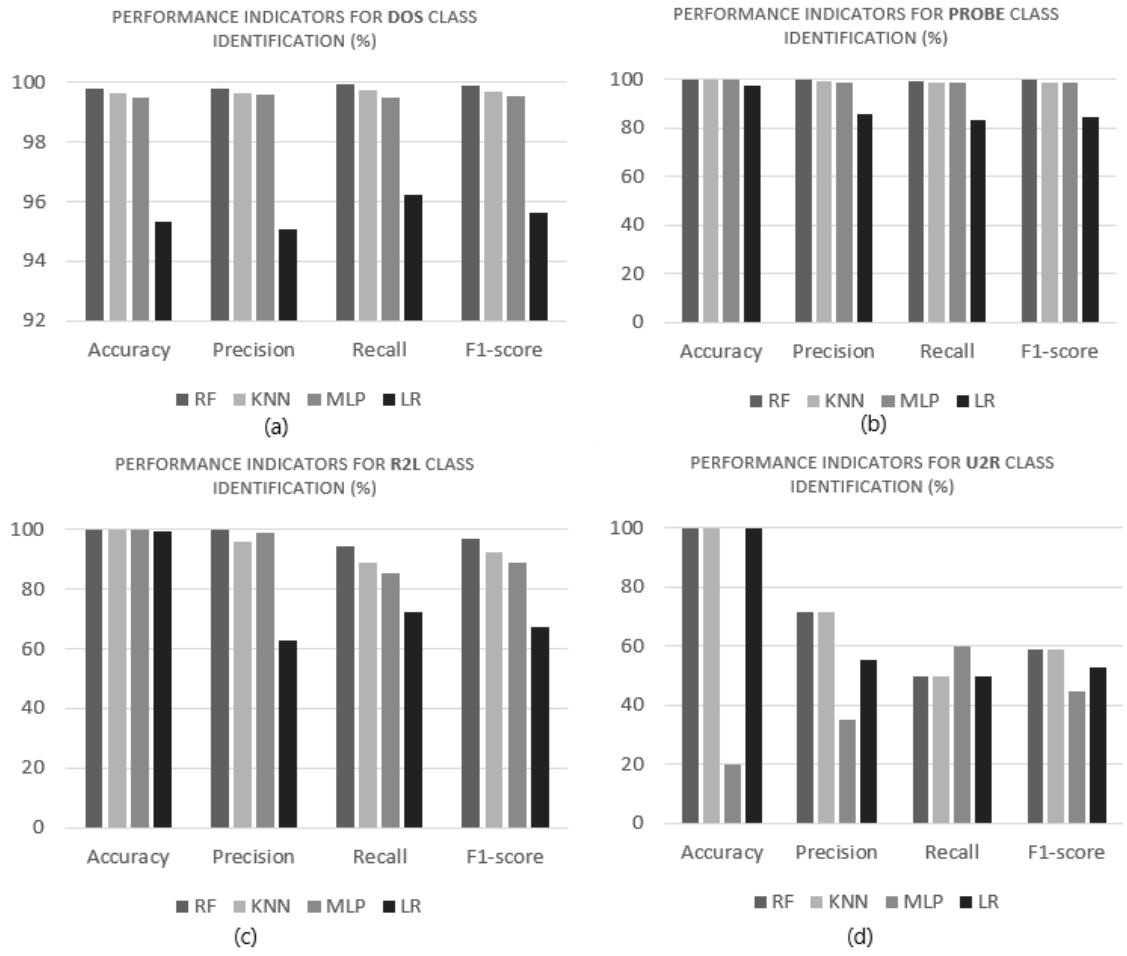


Figure 23. The performance metrics of the proposed models for the 4 attack classes (DoS, Probe, U2R and R2L) identification (%)

Binary classification

As is shown in Figure **24**, the model based on RF shows more effectiveness for the testing accuracy in detecting abnormal and normal behaviors, as it has **99.97%**. Almost all the used algorithms achieved a competitive result since the latter is higher than **99%** for the binary classification. Except for the model based on LR, it has a lower accuracy rate of **96.78%**.

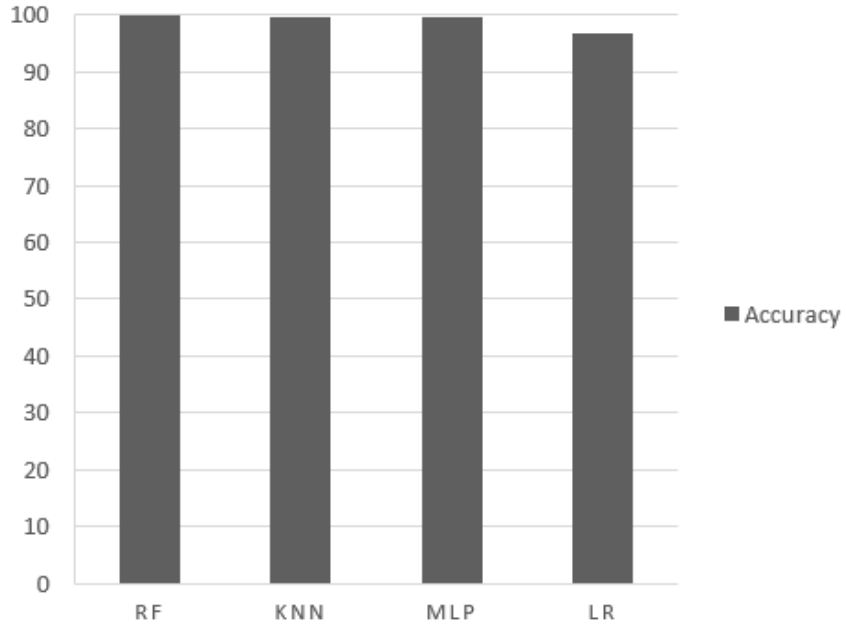


Figure 24. The accuracy rate for testing results for binary classification

Based on the achieved outcome, it is evident that the most appropriate classifier for the proposed **i-2NIDS** model is Random Forest (RF). Thus, it will be called as: **R-2NIDS**. Furthermore, it has been the fastest to make an intelligent decision compared to all other algorithms, as it takes **13.43** seconds.

5.5 Comparison with existing methods

In order to prove the robustness of the proposed model based on Random Forest *(R-2NIDS)*, we have compared the latter to different recent studies in terms of accuracy rate in

distinguishing whether it is a normal or abnormal activity within a network using the NSL-KDD dataset. The figure below depicts the performance comparison.

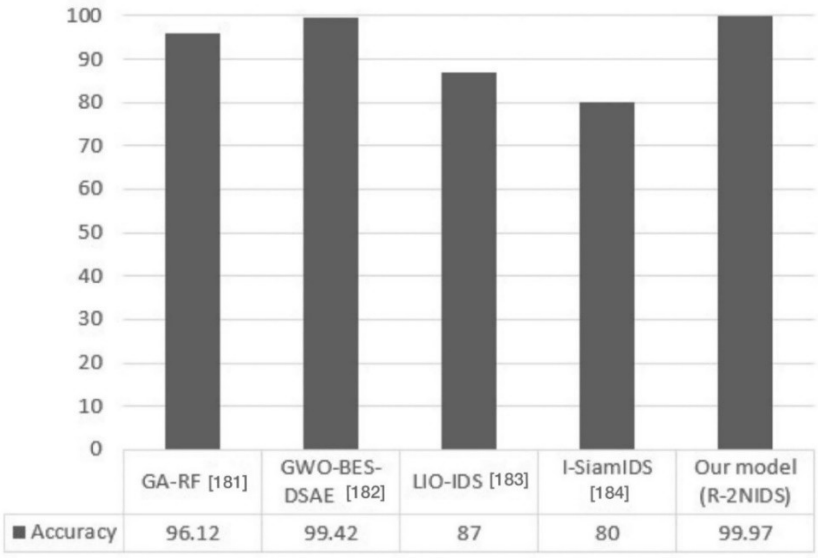


Figure 25. The performance compared with other methods using the NSL-KDD dataset

Figure **25** shows that our proposed model based on RF, which is called R-2NIDS, provides many competitive strong points, compared to other recent techniques, as it has an accuracy of **99.97%**. Whereas in **[181]**, **[182]**, **[183]**, and **[184]** the accuracy of their models reached **96.12%**, **99.42%**, **87%,** and **80%**, respectively. In addition, the proposed approach performs well on imbalanced data and has a better ability to identify different cyberattacks as it has been mentioned in the previous sub-sections.

5.6 Conclusion

An enhanced and efficient Network-based IDS model, called i-2NIDS is presented in this study. The model is proposed based on a procedure that requires splitting the dataset into a train, validation, and test sets over binary and 5 multi-class classification strategies (i.e., Normal class, Dos, Probe, R2L, and U2R attack classes). For the train set, the models based on several machine

learning classifiers (RF, LR, MLP, and KNN) have been trained with different parameters. Then, they are evaluated over the validation set, to be finally tested on the test set with an unbiased estimation of its performance. The model integrated with RF (R-2NIDS) has been selected as the best one in making accurate intelligent decisions with less false alarm rate, using a feature selection approach. It has achieved an accuracy rate of **99.97%** and **99.79%** in detecting normal behaviors and DoS attack class, respectively. The significant improvement of this contribution is, despite the presence of imbalanced network traffic and classification issues existing in the R2L/U2R attack classes, i-2NIDS has the ability to perform well. It obtains an accuracy of **99.76%** and **99.98%** in identifying the R2L and U2R attack classes, respectively. Moreover, compared to other existing approaches that have been recently published, the proposed modeling obtains better results in updating and strengthening the performance of an IDS.

General Conclusion and New Challenges

In conclusion, this endeavor's complexity and importance are highlighted by the objective of optimizing Network Intrusion Detection Systems (NIDS) with the integration of machine learning techniques. It takes careful consideration of data quality, feature selection, model design, training approaches, and evaluation methodologies to improve NIDS performance, which is a multidimensional task. The interplay of these factors, which all affect the system's ability to precisely identify malicious actions inside network data, is crucial to success.

Researchers and practitioners may encourage the development of NIDS solutions that achieve an optimal equilibrium between false positives and false negatives by skillfully establishing the complexities of data preparation, appropriate feature selection, and the selection of relevant algorithms. The necessity of real-time processing, persistent model improvement, and adherence to ethical issues all further emphasize this pursuit's dynamic nature. The resilience to defend digital landscapes against a constantly changing range of threats is ultimately represented in the improvement of NIDS performance through machine learning. For this reason, as a future enhancement of the proposed research studies presented in this report thesis, we will be concentrating more on the prediction time of R-2NIDS, in order to make decisions in real-time. Furthermore, it will be applied to other different IT environments and applications. Overall, this thesis provides interesting findings for the cybersecurity domain. Not only that, but we will also take into consideration the new challenges that machine learning-based intrusion detection systems (IDSs) are facing. The increase in sophisticated and adaptive attack methods emphasizes the necessity for IDSs to accurately identify adversarial attacks, where bad actors use

vulnerabilities in machine learning models to avoid detection. IDSs must respond quickly to new attack patterns because of the difficulty of idea drift introduced by the dynamic nature of cyber threats.

Scientific Publications

Contributions	Title	State
Conference Paper	An Analytical Performance Evaluation of Machine Learning Classifiers for Network Intrusion Detection System	Published
Conference Paper	A Powerful Ensemble Learning Approach for Improving Network Intrusion Detection System	Published
Conference Paper	Importance of Hybridization Modeling for Strengthening the Network Intrusion Detection	Published
Research Paper	i-2NIDS Novel Intelligent Intrusion Detection Approach for a Strong Network Security, Dealing with Imbalanced Classification Issue	Published
Research Paper	Exploration of Adversarial Examples within Network Traffic	Under review

References

[1] Elmaghraby, A. S., & Losavio, M. M. (2014). Cyber security challenges in Smart Cities: Safety, security and privacy. *Journal of advanced research*, *5*(4), 491-497.

[2] Rajasekharaiah, K. M., Dule, C. S., & Sudarshan, E. (2020, December). Cyber security challenges and its emerging trends on latest technologies. In *IOP Conference Series: Materials Science and Engineering* (Vol. 981, No. 2, p. 022062). IOP Publishing.

[3] Rabbani, M., Wang, Y., Khoshkangini, R., Jelodar, H., Zhao, R., Bagheri Baba Ahmadi, S., & Ayobi, S. (2021). A review on machine learning approaches for network malicious behavior detection in emerging technologies. *Entropy*, *23*(5), 529.

[4] Bilge, L., & Dumitraş, T. (2012, October). Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security* (pp. 833-844).

[5] Aludhilu, H., & Rodríguez-Puente, R. (2020). A systematic literature review on intrusion detection approaches. *Revista Cubana de Ciencias Informáticas*, *14*(1), 58-78.

[6] Müller, V. C., & Bostrom, N. (2016). Future progress in artificial intelligence: A survey of expert opinion. *Fundamental issues of artificial intelligence*, 555-572.

[7] Lu, Y. (2019). Artificial intelligence: a survey on evolution, models, applications and future trends. *Journal of Management Analytics*, *6*(1), 1-29.

[8] Kumar, K., & Thakur, G. S. M. (2012). Advanced applications of neural networks and artificial intelligence: A review. *International journal of information technology and computer science*, *4*(6), 57.

[9] Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. *Journal of Industrial Information Integration*, *23*, 100224.

[10] Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. *Machine learning*, 3-23.

[11] Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, 100379.

[12] Li, X., & Shi, Y. (2018, August). Computer vision imaging based on artificial intelligence. In *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)* (pp. 22-25). IEEE.

[13] Wisskirchen, G., Biacabe, B. T., Bormann, U., Muntz, A., Niehaus, G., Soler, G. J., & von Brauchitsch, B. (2017). Artificial intelligence and robotics and their impact on the workplace. *IBA Global Employment Institute*, *11*(5), 49-67.

[14] Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020). How does NLP benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

[15] Chen, S. H., Jakeman, A. J., & Norton, J. P. (2008). Artificial intelligence techniques: an introduction to their use for modelling environmental systems. *Mathematics and computers in simulation*, *78*(2-3), 379-400.

[16] Mukhamediev, R. I., Popova, Y., Kuchin, Y., Zaitseva, E., Kalimoldayev, A., Symagulov, A., ... & Yelis, M. (2022). Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics*, *10*(15), 2552.

[17] Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

[18] Saravanan, R., & Sujatha, P. (2018, June). A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification. In *2018 Second international conference on intelligent computing and control systems (ICICCS)* (pp. 945-949). IEEE.

[19] Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons. b*, *4*, 51-62.

[20] Maulud, D., & Abdulazeez, A. M. (2020). A review on linear regression comprehensive in machine learning. *Journal of Applied Science and Technology Trends*, *1*(4), 140-147.

[21] Rong, S., & Bao-Wen, Z. (2018). The research of regression model in machine learning field. In *MATEC Web of Conferences* (Vol. 176, p. 01033). EDP Sciences.

[22] Fashoto, S. G., Mbunge, E., Ogunleye, G., & den Burg, J. V. (2021). Implementation of machine learning for predicting maize crop yields using multiple linear regression and backward elimination. *Malaysian Journal of Computing (MJoC)*, *6*(1), 679-697.

[23] Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons. b*, *4*, 51-62.

[24] Netrapalli, P. (2019). Stochastic gradient descent and its variants in machine learning. *Journal of the Indian Institute of Science*, *99*(2), 201-213.

[25] Maalouf, M. (2011). Logistic regression in data analysis: an overview. *International Journal of Data Analysis Techniques and Strategies*, *3*(3), 281-299.

[26] Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*, 261-283.

[27] De Ville, B. (2013). Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, *5*(6), 448-455.

[28] Westreich, D., Lessler, J., & Funk, M. J. (2010). Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *Journal of clinical epidemiology*, *63*(8), 826-833.

[29] Lan, T., Hu, H., Jiang, C., Yang, G., & Zhao, Z. (2020). A comparative study of decision tree, random forest, and convolutional neural network for spread-F identification. *Advances in Space Research*, *65*(8), 2052-2061.

[30] Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications*, *134*, 93-101.

[31] Parmar, A., Katariya, R., & Patel, V. (2019). A review on random forest: An ensemble classifier. In *International conference on intelligent data communication technologies and internet of things (ICICI) 2018* (pp. 758-763). Springer International Publishing.

[32] Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, *9*(3), e1301.

[33] Shaik, A. B., & Srinivasan, S. (2019). A brief survey on random forest ensembles in classification model. In *International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2* (pp. 253-260). Springer Singapore.

[34] Chandra, M. A., & Bedi, S. S. (2021). Survey on SVM and their application in image classification. *International Journal of Information Technology*, *13*, 1-11.

[35] Goyal, S. (2022). Effective software defect prediction using support vector machines (SVMs). *International Journal of System Assurance Engineering and Management*, *13*(2), 681-696.

[36] Amato, F., Mazzocca, N., Moscato, F., & Vivenzio, E. (2017, March). Multilayer perceptron: an intelligent model for classification and intrusion detection. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)* (pp. 686-691). IEEE.

[37] Ramchoun, H., Ghanou, Y., Ettaouil, M., & Janati Idrissi, M. A. (2016). Multilayer perceptron: Architecture optimization and training.

[38] Webb, G. I., Keogh, E., & Miikkulainen, R. (2010). Naïve Bayes. *Encyclopedia of machine learning*, *15*(1), 713-714.

[39] Jiang, L., Wang, D., Cai, Z., & Yan, X. (2007). Survey of improving naive bayes for classification. In *Advanced Data Mining and Applications: Third International Conference, ADMA 2007 Harbin, China, August 6-8, 2007. Proceedings 3* (pp. 134-145). Springer Berlin Heidelberg.

[40] Zhu, X. J. (2005). Semi-supervised learning literature survey.

[41] Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine learning*, *109*(2), 373-440.

[42] Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, *12*(3), 729.

[43] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, 485-585.

[44] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, *16*(3), 645-678.

[45] Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, *2*, 165-193.

[46] Na, S., Xumin, L., & Yong, G. (2010, April). Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on intelligent information technology and security informatics* (pp. 63-67). Ieee.

[47] Sinaga, K. P., & Yang, M. S. (2020). Unsupervised K-means clustering algorithm. *IEEE access*, *8*, 80716-80727.

[48] Fodor, I. K. (2002). *A survey of dimension reduction techniques* (No. UCRL-ID-148494). Lawrence Livermore National Lab., CA (US).

[49] Hasan, B. M. S., & Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, *2*(1), 20-30.

[50] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, *7*(3–4), 197-387.

[51] Reddy, S., Allan, S., Coghlan, S., & Cooper, P. (2020). A governance model for the application of AI in health care. *Journal of the American Medical Informatics Association*, *27*(3), 491-497.

[52] Leal Filho, W., Wall, T., Mucova, S. A. R., Nagy, G. J., Balogun, A. L., Luetz, J. M., ... & Gandhi, O. (2022). Deploying artificial intelligence for climate change adaptation. *Technological Forecasting and Social Change*, *180*, 121662.

[53] Cao, L. (2022). Ai in finance: challenges, techniques, and opportunities. *ACM Computing Surveys (CSUR)*, *55*(3), 1-38.

[54] Zhai, X., Chu, X., Chai, C. S., Jong, M. S. Y., Istenic, A., Spector, M., ... & Li, Y. (2021). A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. *Complexity*, *2021*, 1-18.

[55] Michael, K., Abbas, R., Roussos, G., Scornavacca, E., & Fosso-Wamba, S. (2020). Ethics in AI and autonomous system applications design. *IEEE Transactions on Technology and Society*, *1*(3), 114-127.

[56] García-Serrano, A. M., Martínez, P., & Hernández, J. Z. (2004). Using AI techniques to support advanced interaction capabilities in a virtual assistant for e-commerce. *Expert Systems with applications*, *26*(3), 413-426.

[57] Sikos, L. F. (Ed.). (2018). *AI in Cybersecurity* (Vol. 151). Springer.

[58] Thakur, K., Qiu, M., Gai, K., & Ali, M. L. (2015, November). An investigation on cyber security threats and security models. In *2015 IEEE 2nd international conference on cyber security and cloud computing* (pp. 307-311). IEEE.

[59] O'Kane, P., Sezer, S., & Carlin, D. (2018). Evolution of ransomware. *Iet Networks*, *7*(5), 321-327.

[60] Namanya, A. P., Cullen, A., Awan, I. U., & Disso, J. P. (2018, August). The world of malware: An overview. In *2018 IEEE 6th international conference on future internet of things and cloud (FiCloud)* (pp. 420-427). IEEE.

[61] Cheng, L., Liu, F., & Yao, D. (2017). Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(5), e1211.

[62] Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: a literature survey. *IEEE Communications Surveys & Tutorials*, *15*(4), 2091-2121.

[63] Salahdine, F., & Kaabouch, N. (2019). Social engineering attacks: A survey. *Future internet*, *11*(4), 89.

[64] Lau, F., Rubin, S. H., Smith, M. H., & Trajkovic, L. (2000, October). Distributed denial of service attacks. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0* (Vol. 3, pp. 2275-2280). IEEE.

[65] Hunker, J., & Probst, C. W. (2011). Insiders and Insider Threats-An Overview of Definitions and Mitigation Techniques. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, *2*(1), 4-27.

[66] Tatam, M., Shanmugam, B., Azam, S., & Kannoorpatti, K. (2021). A review of threat modelling approaches for APT-style attacks. *Heliyon*, *7*(1).

[67] Ahmad, R., Alsmadi, I., Alhamdani, W., & Tawalbeh, L. A. (2023). Zero-day attack detection: a systematic literature review. *Artificial Intelligence Review*, 1-79.

[68] Butun, I., Österberg, P., & Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. *IEEE Communications Surveys & Tutorials*, *22*(1), 616-644.

[69] Grabe, M., & Christopherson, K. (2005). Evaluating the advantages and disadvantages of providing lecture notes: The role of internet technology as a delivery system and research tool. *The internet and higher education*, *8*(4), 291-298.

[70] Chan, L., Morgan, I., Simon, H., Alshabanat, F., Ober, D., Gentry, J., ... & Cao, R. (2019, June). Survey of AI in cybersecurity for information technology management. In *2019 IEEE technology & engineering management conference (TEMSCON)* (pp. 1-8). IEEE.

[71] Stevens, T. (2018). Global cybersecurity: New directions in theory and methods. *Politics and Governance*, *6*(2), 1-4.

[72] Kim, S. H., Wang, Q. H., & Ullrich, J. B. (2012). A comparative study of cyberattacks. *Communications of the ACM*, *55*(3), 66-73.

[73] Thambiraja, E., Ramesh, G., & Umarani, D. R. (2012). A survey on various most common encryption techniques. *International journal of advanced research in computer science and software engineering*, *2*(7).

[74] May, M. J., Gunter, C. A., & Lee, I. (2006, July). Privacy APIs: Access control techniques to analyze and verify legal privacy policies. In *19th IEEE Computer Security Foundations Workshop (CSFW'06)* (pp. 13-pp). IEEE.

[75] Chen, S., & Song, Q. (2005). Perimeter-based defense against high bandwidth DDoS attacks. *IEEE Transactions on Parallel and Distributed Systems*, *16*(6), 526-537.

[76] Al-Daeef, M. M., Basir, N., & Saudi, M. M. (2017). Security awareness training: A review. *Lecture Notes in Engineering and Computer Science*.

[77] Syed, R. (2020). Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system. *Information & Management*, *57*(6), 103334.

[78] Awuson-David, K., Thompson, J., Tuner, K., & Al-Hadhrami, T. (2021, December). Facilitate security event monitoring and logging of operational technology (OT) legacy systems. In *International Conference of Reliable Information and Communication Technology* (pp. 461-472). Cham: Springer International Publishing.

[79] Thapa, S., & Mailewa, A. (2020, April). The role of intrusion detection/prevention systems in modern computer networks: A review. In *Conference: Midwest Instruction and Computing Symposium (MICS)* (Vol. 53, pp. 1-14)

[80] Sato, M., Yamaki, H., & Takakura, H. (2012, July). Unknown attacks detection using feature extraction from anomaly-based ids alerts. In *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet* (pp. 273-277). IEEE.

[81] Kim, A., Park, M., & Lee, D. H. (2020). AI-IDS: Application of deep learning to real-time Web intrusion detection. *IEEE Access*, *8*, 70245-70261.

[82] Abusitta, A., Bellaiche, M., Dagenais, M., & Halabi, T. (2019). A deep learning approach for proactive multi-cloud cooperative intrusion detection system. *Future Generation Computer Systems*, *98*, 308-318.

[83] Sowmya, T., & Anita, E. M. (2023). A comprehensive review of AI based intrusion detection system. *Measurement: Sensors*, 100827.

[84] Nguyen, H. T., & Franke, K. (2012, December). Adaptive Intrusion Detection System via online machine learning. In *2012 12th international conference on hybrid intelligent systems (HIS)* (pp. 271-277). IEEE.

[85] Sawant, A. (2018, August). A comparative study of different intrusion prevention systems. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* (pp. 1-5). IEEE.

[86] Bhati, N. S., Khari, M., García-Díaz, V., & Verdú, E. (2020). A review on intrusion detection systems and techniques. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *28*(Supp02), 65-91.

[87] Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy.

[88] Brown, D. J., Suckow, B., & Wang, T. (2002). A survey of intrusion detection systems. *Department of Computer Science, University of California, San Diego*.

[89] Ashoor, A. S., & Gore, S. (2011). Importance of intrusion detection system (IDS). *International Journal of Scientific and Engineering Research*, *2*(1), 1-4.

[90] Hoque, M. S., Mukit, M. A., & Bikas, M. A. N. (2012). An implementation of intrusion detection system using genetic algorithm. *arXiv preprint arXiv:1204.1336*.

[91] Kumar, V., & Sangwan, O. P. (2012). Signature based intrusion detection system using SNORT. *International Journal of Computer Applications & Information Technology*, *1*(3), 35-41.

[92] Jyothsna, V. V. R. P. V., Prasad, R., & Prasad, K. M. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, *28*(7), 26-35.

[93] Kwon, H. Y., Kim, T., & Lee, M. K. (2022). Advanced intrusion detection combining signature-based and behavior-based detection methods. *Electronics*, *11*(6), 867.

[94] Rupa Devi, T., & Badugu, S. (2020). A review on network intrusion detection system using machine learning. In *Advances in Decision Sciences, Image Processing, Security and Computer Vision: International Conference on Emerging Trends in Engineering (ICETE), Vol. 2* (pp. 598-607). Springer International Publishing.

[95] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.

[96] Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018, April). A survey on anomaly based host intrusion detection system. In *Journal of Physics: Conference Series* (Vol. 1000, p. 012049). IOP Publishing.

[97] Liu, M., Xue, Z., Xu, X., Zhong, C., & Chen, J. (2018). Host-based intrusion detection system with system calls: Review and future trends. *ACM Computing Surveys (CSUR)*, *51*(5), 1-36.

[98] Modi, C., Patel, D., Borisaniya, B., Patel, H., Patel, A., & Rajarajan, M. (2013). A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, *36*(1), 42-57.

[99] Ho, C. Y., Lai, Y. C., Chen, I. W., Wang, F. Y., & Tai, W. H. (2012). Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems. *IEEE Communications Magazine*, *50*(3), 146-154.

[100] Asgharzadeh, H., Ghaffari, A., Masdari, M., & Gharehchopogh, F. S. (2023). Anomaly-based intrusion detection system in the Internet of Things using a convolutional neural network and multi-objective enhanced Capuchin Search Algorithm. *Journal of Parallel and Distributed Computing*, *175*, 1-21.

[101] Li, W., Meng, W., & Au, M. H. (2020). Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in IoT environments. *Journal of Network and Computer Applications*, *161*, 102631.

[102] Hubballi, N., Biswas, S., & Nandi, S. (2011). Network specific false alarm reduction in intrusion detection system. *Security and Communication Networks*, *4*(11), 1339-1349.

[103] Idrissi, I., Boukabous, M., Azizi, M., Moussaoui, O., & El Fadili, H. (2021). Toward a deep learning-based intrusion detection system for IoT against botnet attacks. *IAES International Journal of Artificial Intelligence*, *10*(1), 110.

[104] Beaver, J. M., Borges-Hink, R. C., & Buckner, M. A. (2013, December). An evaluation of machine learning methods to detect malicious SCADA communications. In *2013 12th international conference on machine learning and applications* (Vol. 2, pp. 54-59). IEEE.

[105] Kaur, H., Singh, G., & Minhas, J. (2013). A review of machine learning based anomaly detection techniques. *arXiv preprint arXiv:1307.7286*.

[106] Pietraszek, T., & Tanner, A. (2005). Data mining and machine learning—towards reducing false positives in intrusion detection. *Information security technical report*, *10*(3), 169-183.

[107] Papamartzivanos, D., Mármol, F. G., & Kambourakis, G. (2019). Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE access*, *7*, 13546-13560.

[108] Azeroual, O., & Nikiforova, A. (2022). Apache spark and mllib-based intrusion detection system or how the big data technologies can secure the data. *Information*, *13*(2), 58.

[109] Felix, E. A., & Lee, S. P. (2019). Systematic literature review of preprocessing techniques for imbalanced data. *IET Software*, *13*(6), 479-496.

[110] Jain, A., Patel, H., Nagalapatti, L., Gupta, N., Mehta, S., Guttula, S., ... & Munigala, V. (2020, August). Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3561-3562).

[111] Roelofs, R. (2019). *Measuring Generalization and overfitting in Machine learning*. University of California, Berkeley.

[112] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

[113] Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (Eds.). (2008). *Feature extraction: foundations and applications* (Vol. 207). Springer.

[114] Sangkatsanee, P., Wattanapongsakorn, N., & Charnsripinyo, C. (2011). Practical real-time intrusion detection using machine learning approaches. *Computer Communications*, *34*(18), 2227-2235.

[115] Othman, S. M., Alsohybe, N. T., Ba-Alwi, F. M., & Zahary, A. T. (2018). Survey on intrusion detection system types. *International Journal of Cyber-Security and Digital Forensics*, *7*(4), 444-463.

[116] Can, O., & Sahingoz, O. K. (2015, May). A survey of intrusion detection systems in wireless sensor networks. In *2015 6th international conference on modeling, simulation, and applied optimization (ICMSAO)* (pp. 1-6). IEEE.

[117] Ray, L. L. (2013). Training and testing anomaly-based neural network intrusion detection systems. *International Journal of Information Security Science*, *2*(2), 57-63.

[118] Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. A. (2020). Performance analysis of machine learning algorithms in intrusion detection system: A review. *Procedia Computer Science*, *171*, 1251-1260.

[119] Ennaji, S., Haddouch, K., & El Akkad. N. (2021). An Analytical Performance Evaluation of Machine Learning Classifiers for Network Intrusion Detection System. *International Conference on Embedded Systems and Artificial Intelligence.* Springer.

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

[121] Abirami, M. S., Yash, U., & Singh, S. (2020). Building an ensemble learning based algorithm for improving intrusion detection system. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 635-649). Springer, Singapore.

[122] Polikar, R. (2012). Ensemble learning. In *Ensemble machine learning* (pp. 1-34). Springer, Boston, MA.

[123] Yihunie, F., Abdelfattah, E., & Regmi, A. (2019, May). Applying machine learning to anomaly-based intrusion detection systems. In *2019 IEEE Long Island Systems, Applications and Technology Conference (LISAT)* (pp. 1-5). IEEE.

[124] Gu, J., Wang, L., Wang, H., & Wang, S. (2019). A novel approach to intrusion detection using SVM ensemble with feature augmentation. *Computers & Security*, *86*, 53-62.

[125] Hsu, Y. F., He, Z., Tarutani, Y., & Matsuoka, M. (2019, July). Toward an online network intrusion detection system based on ensemble learning. In *2019 IEEE 12th international conference on cloud computing (CLOUD)* (pp. 174-178). IEEE.

[126] Ludwig, S. A. (2017, November). Intrusion detection of multiple attack classes using a deep neural net ensemble. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-7). IEEE.

[127] Rajadurai, H., & Gandhi, U. D. (2020). A stacked ensemble learning model for intrusion detection in wireless network. *Neural computing and applications*, 1-9.

[128] Haq, N. F., Onik, A. R., & Shah, F. M. (2015, November). An ensemble framework of anomaly detection using hybridized feature selection approach (HFSA). In *2015 SAI Intelligent Systems Conference (IntelliSys)* (pp. 989-995). IEEE.

[129] Tama, B. A., Comuzzi, M., & Rhee, K. H. (2019). TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. *IEEE Access*, *7*, 94497-94507.

[130] Das, S., Mahfouz, A. M., Venugopal, D., & Shiva, S. (2019, July). DDoS intrusion detection through machine learning ensemble. In *2019 IEEE 19th international conference on software Quality, Reliability and Security Companion (QRS-C)* (pp. 471-477). IEEE.-

[131] Khonde, S. R., & Ulagamuthalvi, V. (2019). Ensemble-based semi-supervised learning approach for a distributed intrusion detection system. *Journal of Cyber Security Technology*, *3*(3), 163-188.

[132] Ahmad, I., Basheri, M., Iqbal, M. J., & Rahim, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. *IEEE access*, *6*, 33789-33795.

[133] Ravipati, R. D., & Abualkibash, M. (2019). Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets-a review paper. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, *11*.

[134] Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. *Procedia Computer Science*, *89*, 213-217.

[135] Hasan, M. A. M., Nasser, M., Ahmad, S., & Molla, K. I. (2016). Feature selection for intrusion detection using random forest. *Journal of information security*, *7*(3), 129-140.

[136] Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2014). Support vector machine and random forest modeling for intrusion detection system (IDS). *Journal of Intelligent Learning Systems and Applications*, *2014*.

[137] Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. *Procedia Computer Science*, *89*, 117-123.

[138] Bisong, E. (2019). *Building machine learning and deep learning models on Google Cloud Platform* (pp. 7-10). Berkeley: Apress.

[139] Panda, M., & Patra, M. R. (2007). Network intrusion detection using naive bayes. *International journal of computer science and network security*, *7*(12), 258-263.

[140] Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. (2022). Comparative research on network intrusion detection methods based on machine learning. *Computers & Security*, 102861.

[141] Saheed, Y. K., Arowolo, M. O., & Tosho, A. U. (2022). An efficient hybridization of k-means and genetic algorithm based on support vector machine for cyber intrusion detection system. *International Journal on Electrical Engineering and Informatics*, *14*(2), 426-442.

[142] Alkanhel, R., Khafaga, D. S., El-kenawy, E. S. M., Abdelhamid, A. A., Ibrahim, A., Amin, R., ... & El-den, B. M. (2023). Hybrid Grey Wolf and Dipper Throated Optimization in Network Intrusion Detection Systems. *CMC-COMPUTERS MATERIALS & CONTINUA*, *74*(2), 2695-2709.

[143] Prakash, P. J., & Lalitha, B. (2022). Optimized ensemble classifier based network intrusion detection system for RPL based internet of things. *Wireless Personal Communications*, *125*(4), 3603-3626.

[144] Jahani, Babak, and Babak Mohammadi. "A comparison between the application of empirical and ANN methods for estimation of daily global solar radiation in Iran." Theoretical and Applied Climatology 137.1-2 (2019): 1257-1269.

[145] Tang, Peng, et al. "Detection of SQL injection based on artificial neural network." Knowledge-Based Systems 190 (2020): 105528.

[146] A. Shenfield, D. Day, A. Ayesh, Intelligent intrusion detection systems using artificial neural networks, ICT Express (2018)

[147] Panda, M., Abraham, A., & Patra, M. R. (2010, August). Discriminative multinomial naive bayes for network intrusion detection. In 2010 Sixth International Conference on Information Assurance and Security (pp. 5-10). IEEE.

[148] Sravani, K., & Srinivasu, P. (2014). Comparative study of machine learning algorithm for intrusion detection system. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013 (pp. 189-196). Springer, Cham.

[149] Gao, Xianwei, et al. "An adaptive ensemble machine learning model for intrusion detection." *IEEE Access* 7 (2019): 82512-82521.

[150] Abirami, M. S., Umaretiya Yash, and Sonal Singh. "Building an ensemble learning based algorithm for improving intrusion detection system." *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, Singapore, 2020. 635-649.

[151] Khonde, S. R., and V. Ulagamuthalvi. "Ensemble-based semi-supervised learning approach for a distributed intrusion detection system." *Journal of Cyber Security Technology* 3.3 (2019): 163-188.

[152] Das, Saikat, et al. "DDoS intrusion detection through machine learning ensemble." *2019 IEEE 19th international conference on software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2019.

[153] Tama, Bayu Adhi, Marco Comuzzi, and Kyung-Hyune Rhee. "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system." *IEEE Access* 7 (2019): 94497-94507.

[154] Polikar, Robi. "Ensemble learning." *Ensemble machine learning*. Springer, Boston, MA, 2012. 1-34.

[155] Lian, Wenjuan, et al. "An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning." *Mathematical Problems in Engineering* 2020 (2020).

[156] AlShahrani, Bdah Mohammed Mubarak. "Classification of Cyber-Attack using Adaboost Regression Classifier and Securing the Network." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.10 (2021): 1215-1223.

[157] Paliwal, S., & Gupta, R. (2012). Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm. International Journal of Computer Applications, 60(19), 57-62.

[158] Ennaji, S., El Akkad, N., & Haddouch, K. (2023). i-2NIDS Novel Intelligent Intrusion Detection Approach for a Strong Network Security. *International Journal of Information Security and Privacy (IJISP)*, *17*(1), 1-17.

[159] Rosay, A., Carlier, F., & Leroux, P. (2020). MLP4NIDS: An efficient MLP-Based network intrusion detection for CICIDS2017 dataset. In *Machine Learning for Networking: Second IFIP TC 6 International Conference, MLN 2019, Paris, France, December 3–5, 2019, Revised Selected Papers 2* (pp. 240-254). Springer International Publishing.

[160] Schapire, Robert E. "Explaining adaboost." *Empirical inference*. Springer, Berlin, Heidelberg, 2013. 37-52.

[161] Hosseini, Soodeh, and Behnam Mohammad Hasani Zade. "New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN." *Computer Networks* 173 (2020): 107168.

[162] Golrang, Anahita, et al. "A novel hybrid IDS based on modified NSGAII-ANN and random forest." *Electronics* 9.4 (2020): 577.

[163] Meryem, Amar, and Bouabid EL Ouahidi. "Hybrid intrusion detection system using machine learning." *Network Security* 2020.5 (2020): 8-19.

[164] Samriya, Jitendra Kumar, and Narander Kumar. "A novel intrusion detection system using hybrid clustering-optimization approach in cloud computing." *Materials Today: Proceedings* (2020).

[165] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, *36*(2), 111-133.

[166] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009, July). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications* (pp. 1-6). IEEE.

[167] Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural computation*, *9*(7), 1545-1588.

[168] Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[169] Resende, P. A. A., & Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)*, *51*(3), 1-36.

[170] Gladence, L. M., Karthi, M., & Anu, V. M. (2015). A statistical comparison of logistic regression and different Bayes classification methods for machine learning. *ARPN Journal of Engineering and Applied Sciences*, *10*(14), 5947-5953.

[171] Faizal, M. A., Sahib, S., Robiah, Y., & Rahayu, S. (2010, September). Time based intrusion detection on fast attack for network intrusion detection system. In *2010 Second International Conference on Network Applications, Protocols and Services* (pp. 148-152). IEEE.

[172] Sheta, A. F., & Alamleh, A. (2015, December). A professional comparison of c4. 5, mlp, svm for network intrusion detection based feature analysis. In *The International Congress for global Science and Technology* (Vol. 47, p. 15).

[173] Khater, B. S., Abdul Wahab, A. W., Idris, M. Y. I., Hussain, M. A., Ibrahim, A. A., Amin, M. A., & Shehadeh, H. A. (2021). Classifier performance evaluation for lightweight IDS using fog computing in IoT security. *Electronics*, *10*(14), 1633.

[174] Arora, K., & Chauhan, R. (2017, April). Improvement in the performance of deep neural network model using learning rate. In *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)* (pp. 1-5). IEEE.

[175] Tang, P., Qiu, W., Huang, Z., Lian, H., & Liu, G. (2020). Detection of SQL injection based on artificial neural network. *Knowledge-Based Systems*, *190*, 105528.

[176] Lubis, A. R., & Lubis, M. (2020). Optimization of distance formula in K-Nearest Neighbor method. *Bulletin of Electrical Engineering and Informatics*, *9*(1), 326-338.

[177] Kumar, I., Mohd, N., Bhatt, C., & Sharma, S. K. (2020). Development of IDS using supervised machine learning. In *Soft computing: Theories and applications* (pp. 565-577). Springer, Singapore.

[178] Wazirali, R. (2020). An Improved Intrusion Detection System Based on KNN Hyperparameter Tuning and Cross-Validation. *Arabian Journal for Science and Engineering*, *45*(12), 10859-10873.

[179] Bilbao, I., & Bilbao, J. (2017, December). Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks. In *2017 eighth international conference on intelligent computing and information systems (ICICIS)* (pp. 173-177). IEEE.

[180] Mendez, K. M., Pritchard, L., Reinke, S. N., & Broadhurst, D. I. (2019). Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. *Metabolomics*, *15*(10), 1-16.

[181] Liu, Z., & Shi, Y. (2022). A Hybrid IDS Using GA-Based Feature Selection Method and Random Forest. *International Journal of Machine Learning and Computing*, *12*(2).

[182] Keserwani, P. K., Govil, M. C., Pilli, E. S., & Govil, P. (2021). An Optimal NIDS for VCN Using Feature Selection and Deep Learning Technique: IDS for VCN. *International Journal of Digital Crime and Forensics (IJDCF)*, *13*(6), 1-25.

[183] Gupta, N., Jindal, V., & Bedi, P. (2021). LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system. *Computer Networks*, *192*, 108076.

[184] Bedi, P., Gupta, N., & Jindal, V. (2021). I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems. *Applied Intelligence*, *51*(2), 1133-1151.