Saad Dahleb University of Blida 1

Faculty of Sciences

Computer Science department

Thesis presented by:

- Mr.Atsamnia Youcef
- Mr.Boucheloukh Naaman

For obtaining the Master's Degree

Domaine : Mathematics and Computer Science

Major : Computer Science

Specialty : Computer system and network

---

eXplainable Artificial Intelligence for Intrusion Detection System based on ML

Defended on 02/07/2024, in front of the jury composed of

President	Ms. Boumahdi
Examiner	Ms. Zahra
Supervisor	Ms.Boustia Narimane
Co-Supervisor	Mr SI Ahmed Ayoub

Acknowledgment

First and foremost, I thank ALLAH for granting me health, patience, energy, and the will to complete this work.

I wish to thank the Head of the Computer Science Department for their constant advice and support.

I would like to express my sincere appreciation to our supervisor, Professor Ms. Boustia Narimane, for her continuous guidance, support, and encouragement throughout my Masters at Blida University. I am also grateful to Mr. Si Ahmed Ayoub for his expertise, mentorship, and patience, which have significantly contributed to the completion of this thesis. I extend my gratitude to all our teachers who contributed to our education, all the teachers in the department, and especially to the members of the jury. A special thanks to my thesis mate, Mr. Atsamnia Youcef, for his collaboration, support, and teamwork throughout this journey.

Finally, I would like to thank my parents for their unwavering support throughout my academic journey. Their love and support have been a source of strength and motivation, without which this accomplishment would not have been possible.

I dedicate this work to:

My dear wife, who has been my steadfast support and source of strength, always lighting up my life with her deep love and unwavering hope.

My sons, Ghaith and Abderrahmane, for their smiles and joy.

My brothers and sister, who have shared both my joys and challenges.

All my colleagues and professors, who have guided me in my academic journey.

My friends, with whom I have shared the experiences of my scientific journey and my work colleagues.

All those who are close to my heart.

NAAMAN

First and foremost, we thank and praise ALLAH, the Almighty, for assisting and taking care of us in each step we made along the path despite the wrong decisions we make, and providing us with the strength and ability to complete this thesis. I would also like to extend my heartfelt thanks to my supervisor, Pf. Ms. Bousita Narimane, and my co-supervisor, Mr. SI-Ahmed Ayoub. Your guidance, patience, and invaluable insights have been instrumental in shaping this thesis. my gratitude to all our teachers who con- tributed to our education your dedication to my success has been truly inspiring, and I am deeply appreciative of your mentorship.

I dedicate this work To my beloved parents, for their unwavering support and endless encouragement.To my brother Mohamed keeping me always aiger for working a and motivated, Abdelwahab Sharing my hard times with me, my sisters and there kids keeping the smile and the joy on my face, thank you for your constant support and for always being there to cheer me on. Your encouragement has been a source of strength, and I am grateful for the bond we share.

A heartfelt thanks to Mr. Haireche Sofiane for consistently encouraging and motivating me to delve deeper into various fields of knowledge, continually pushing me to learn and grow. Your support has been invaluable in expanding my understanding across multiple domains.

I am also incredibly grateful to my friends Younes, Azzedine and Amani for their support and camaraderie. Thank you for moments of laughter that made this journey enjoyable. Your friendship has been a vital part of my life, and I cherish the memories we have created together.

To my thesis mate, Naaman, thank you for your collaboration, support, and the shared dedication to our research. Working alongside you has been a rewarding experience, and your insights and companionship have greatly enriched this project. Lastly, I would like to thank everyone who has contributed to this thesis in any way. Your support and encouragement have made this accomplishment possible, and I am deeply appreciative of each and every one of you.

YOUCEF

Abstract

The advancement of technology has reshaped various domains, particularly cybersecurity, where increasingly sophisticated cyberattacks pose significant threats. Explainable Artificial Intelligence (XAI) addresses the crucial need for transparency in AI systems. This thesis investigates XAI's application to Intrusion Detection Systems (IDS) using Machine Learning (ML) and Deep Learning (DL) on Network-based (NIDS) and Host-based (HIDS) datasets. Specifically, the study utilizes the UNSW-NB15 and CIC-IDS2018 datasets to evaluate the performance of Artificial Neural Networks (ANN) and XGBoost algorithms. Both algorithms have demonstrated significant effectiveness, achieving high accuracy and robust detection capabilities in identifying various types of cyberattacks.

The thesis further explores the use of local agnostic LIME (Local Interpretable Model-agnostic Explanations) and global agnostic SHAP (SHapley Additive exPlanations) to enhance the interpretability of AI models. These XAI methods provide detailed insights into model decisions, making the AI-driven processes more transparent and trustworthy. Empirical evidence shows that LIME and SHAP not only improve the understanding of model behavior but also highlight the strengths and weaknesses of ANN and XGBoost in different scenarios.

The study offers valuable insights for cybersecurity professionals and policymakers, demonstrating that the integration of AI, ML, DL, IDS, and XAI can significantly improve cybersecurity by making AI-driven decisions more transparent and trustworthy. These findings underscore the potential of combining advanced algorithms with interpretability techniques to develop more reliable and effective intrusion detection systems.

Keywords:

Explainable Artificial Intelligence (XAI), Intrusion Detection System (IDS), Machine

Learning (ML), Deep Learning (DL), Network Intrusion Detection System (NIDS), Host-based Intrusion Detection System (HIDS), Local Interpretable Model-agnostic Explanations (LIME), and SHapley Additive exPlanations (SHAP).

Résumé

L'avancement de la technologie a remodelé divers domaines, en particulier la cybersécurité, où des cyberattaques de plus en plus sophistiquées posent des menaces significatives. L'Intelligence Artificielle Explicable (XAI) répond au besoin crucial de transparence dans les systèmes d'IA. Cette thèse examine l'application de la XAI aux systèmes de détection d'intrusion (IDS) en utilisant l'apprentissage automatique (ML) et l'apprentissage profond (DL) sur des ensembles de données basés sur le réseau (NIDS) et basés sur l'hôte (HIDS). Plus précisément, l'étude utilise les ensembles de données UNSW-NB15 et CIC-IDS2018 pour évaluer les performances des algorithmes de réseaux de neurones artificiels (ANN) et XGBoost. Les deux algorithmes ont démontré une efficacité significative, atteignant une grande précision et des capacités de détection robustes pour identifier divers types de cyberattaques.

La thèse explore en outre l'utilisation de LIME (Local Interpretable Model-agnostic Explanations) agnostique local et de SHAP (SHapley Additive exPlanations) agnostique global pour améliorer l'interprétabilité des modèles d'IA. Ces méthodes XAI fournissent des informations détaillées sur les décisions des modèles, rendant les processus pilotés par l'IA plus transparents et dignes de confiance. Les preuves empiriques montrent que LIME et SHAP améliorent non seulement la compréhension du comportement des modèles, mais mettent également en évidence les forces et les faiblesses des ANN et XGBoost dans différents scénarios.

L'étude offre des informations précieuses pour les professionnels de la cybersécurité et les décideurs, démontrant que l'intégration de l'IA, ML, DL, IDS et XAI peut améliorer considérablement la cybersécurité en rendant les décisions pilotées par l'IA plus transparentes et dignes de confiance. Ces résultats soulignent le potentiel de la combinaison d'algorithmes avancés avec des techniques d'interprétabilité pour développer des systèmes de détection d'intrusion plus fiables et efficaces.

Mots clés :

intelligence Artificielle Explicable (XAI), Système de Détection d'Intrusion (IDS), Apprentissage Automatique (ML), Apprentissage Profond (DL), Système de Détection d'Intrusion Réseau (NIDS), Système de Détection d'Intrusion Basé sur l'Hôte (HIDS), Explications Modèles Indépendants et Localement Interprétables (LIME), et Explications Additives de Shapley (SHAP).

Contents

List of Tables

List of Figures

Acronyms

.

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

IDS Intrusion Detection System

ANN Artificial Neural Network

XGBoost eXtreme Gradient Boosting

CNN Convolutional Neural Network

IoMT Internet of Medical Things

XAI eXplainable Artificial Intelligence

DL Deep Learning

HIDS Host-based Intrusion Detection System

NIDS Network-based Intrusion Detection System

SHAP SHapley Additive exPlanations

LIME Local Interpretable Model-Agnostic Explanations

DoS Denial of Service

DDoS Distributed Denial of Service

MITM Man-in-the-Middle

DMZ Demilitarized Zone

General Introduction

The rapid advancement of technology has significantly reshaped the landscape of various domains, particularly in the realm of cybersecurity. As digital transformation continues to accelerate, the sophistication and frequency of cyberattacks have also increased, posing substantial threats to individuals, organizations, and governments worldwide. AI is considered a vital tool to mitigate these evolving cyber threats. The capability of AI to process and analyze vast amounts of data enables it to detect anomalies that may indicate zero-day attacks and new vulnerabilities. Zero-day attacks exploit previously unknown vulnerabilities and are particularly challenging to identify with traditional methods. AI can recognize patterns and anomalies that suggest such attacks, providing a crucial layer of defense.

Although the term Explainable Artificial Intelligence (XAI) has gained popularity only recently, it is a concept with roots dating back several decades [2], to the early stages of Artificial Intelligence (AI) and ML . In the last decade, the widespread availability of cheap computational power and abundant data has propelled AI and ML into nearly every domain. These advancements have led AI to achieve (super) human performance in various tasks, driving its adoption in , law, defense, finance, self-driving cars and healthcare.

For exemple in healthcare sector integration of the Internet of Things (IoT) has significantly enhanced the efficiency and effectiveness of medical services. By 2023, over 161 million IoT devices are expected to be in use in hospitals, ranging from patient monitors to imaging systems. However, IoT systems suffer from significant issues, particularly cybersecurity attacks. To mitigate these issues, IDS based on ML are crucial. For cybersecurity experts to trust the decisions taken by IDS, it is essential to understand how these decisions are made. XAI methods address this problem by making the decision-making processes of AI systems more transparent, turning black-box models into more interpretable.

In this thesis, the application of XAI to IDS using two complex models with highest accuracy, ML and DL are explored on two different types of datasets one based on NIDS and the other on HIDS, aiming to improve the detection of sophisticated

cyberattacks while maintaining transparency and trustworthiness. The primary objectives are to evaluate the effectiveness of XAI method local agnostic LIME and global agnostic SHAP in making AI models more interpretable and to assess how this impacts performance and trust. The contrubutions of thes study is providing empirical evidence on the benefits and challenges of implementing XAI methods for IDS, offering valuable insights for cybersecurity professionals, researchers, and policymakers.

This thesis is organized into three chapters. Chapter I provides background and related work. Chapter II details the research methodology used in the study. Chapter III implementation and discusses the results. Finally, the general conclusion summarizes the research, limitations, and recommendations for future study. .

Chapter I

BACKGROUND

I.1 Introduction

The rapid advancement of AI brings significant challenges, especially the opacity and inscrutability of AI models, which are critical in domains like cybersecurity. To address these challenges, Explainable AI (XAI) aims to clarify AI decision-making processes, enhancing understanding and trust.

This chapter employs thematic analysis to examine various XAI approaches within the context of intrusion detection, with a particular focus on LIME and SHAP, two of the most prominent methods in the field. By exploring the foundational concepts of XAI and its application to IDS, this chapter aims to illuminate the path towards more transparent, accountable, and effective cybersecurity solutions.

The chapter also initiates a comprehensive review of the extant literature on XAI and IDS, highlighting that ML-based IDS offer a self-learning solution and outperform traditional IDS. As AI continues to develop, the need for open and understandable models becomes increasingly apparent, especially in crucial domains such as cybersecurity. Understanding how AI models make decisions is essential for maintaining the reliability, accountability, and trustworthiness of automated decision-making systems.

I.2 Definitions

I.2.1 Artificial Intelligence

Artificial Intelligence (AI) is a revolutionary computer program designed to mimic human cognition and behavior, thereby enhancing our lives and productivity. Emerging at a time of improving living standards, AI with the use of AI models offers unparalleled convenience and benefits to humanity [3]. This branch of computer science aims to delve into the intricacies of intelligence, paving the way for the creation of intelligent machines capable of learning, reasoning, and adapting to their environment. With applications spanning robotics, language and image recognition, natural

language processing, and more, AI continues to evolve, striving to surpass human cognitive capabilities and redefine the boundaries of technological advancement [4].

I.2.2 Machine Learning

Machine Learning (ML) answers the question of how to build computers that automatically improve with experience. It is one of the fastest growing technical fields today, located at the intersection of computer science and statistics, and at the core of AI and data science. Recent advances in ML have been spurred both by the development of new and theoretical learning algorithms and by the availability of online data and low-cost computations. The adoption of data-intensive ML methods can be found in all areas of science, technology and business, leading to more evidence-based decision-making in many areas of life [5].

Types of ML

ML algorithms are organized based on the desired outcomes, with one major category being classification. Common types of ML algorithms include [6]:

1. **Supervised learning :** Where the algorithm generates a function that maps inputs to desired outputs using labeled examples It is used to create predictive models that can be used to make decisions and predictions [7].

2. **Unsupervised learning:** Which involves training algorithms to identify patterns and relationships in data without the use of labels or prior knowledge. It is used to explore and analyze data to discover hidden patterns and correlations [8].

3. **Semi-supervised learning:** Which combines both labeled and unlabeled examples to generate an appropriate function or classifier that can learn from both labeled and unlabeled data and improve their accuracy [9].

4. **Reinforcement learning:** Reinforcement learning is a type of ML in which an agent learns to make decisions

   by interacting with its environment and receiving feedback in the form of rewards or penalties [10].

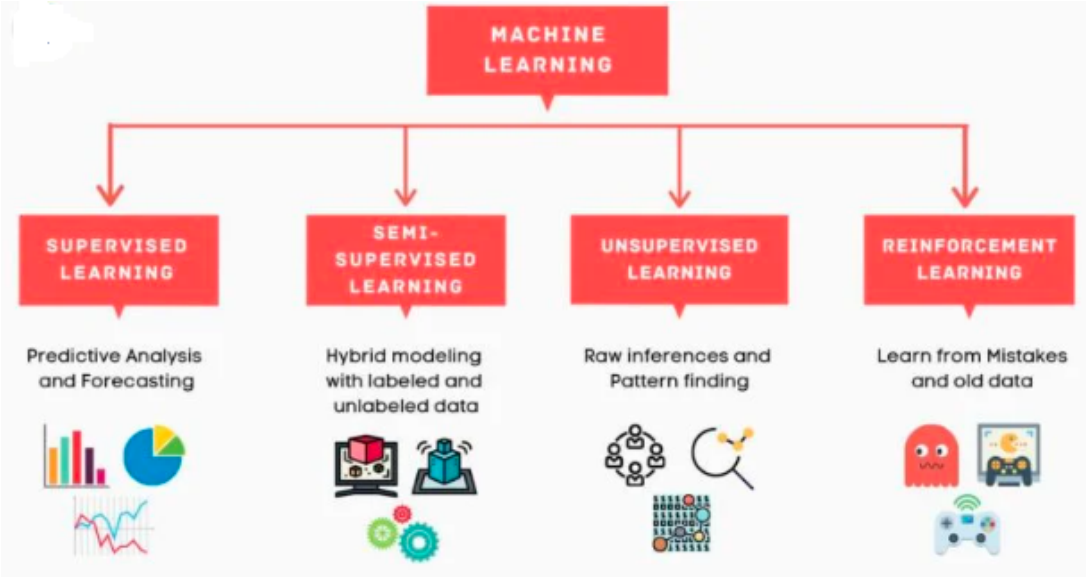


Figure I.1: Types of ML [1]

I.2.3 Deep Learning

DL is a branch of ML characterized by its focus on learning hierarchical representations of data through multiple levels of abstraction and feature extraction. These representations, formed through cascades of nonlinear processing units across layers, do not capture the semantics of real-world problems. However, the number of layers may have the semantics; for example, only more than one hidden layer can represent a non-linear classification relation between inputs and outputs [11], aim to capture complex relationships and patterns within the input data. While DL algorithms do not inherently assign semantics to individual layers, the number and arrangement of layers can encode meaningful information, such as non-linear classification relations. DL models prioritize learning data representations over task-specific algorithms and

can operate in supervised, semi-supervised, or unsupervised learning settings. Despite variations in specialized methods like Artificial Neural Networks (ANN), DL serves as a powerful tool for uncovering intricate structures and insights from diverse data modalities, such as sound, image, and text.

I.2.4 eXplainable Artificial Intelligence

The term "eXplainable Artificial Intelligence (XAI)" refers to a branch of research and development that aims at improving transparency and intelligibility of AI systems a branch of study that focuses on ML interpretability techniques. It was once overlooked in favor of “black box” [12] models optimized for performance until AI applications became complex and had societal effects. The drivers behind XAI are primarily ethics, responsibility, and equity in the sense that the goal is to make transparent the decision-making process of AI models so as to regain confidence among human users. Main goals include among others ensuring balancedness, recognizing mistakes, enabling collaboration between people and machines, plus making AI results more understandable. Various methods exist such as model-based approaches, post-hoc explainability which include two important models (Black box/White box),

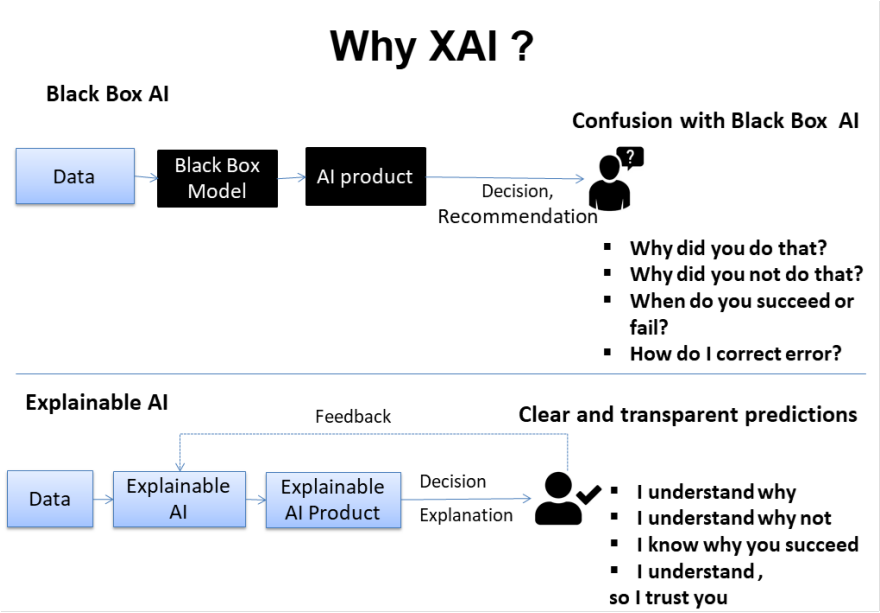


Figure I.2: The Importance of Explainable AI

White box models	Black box models
Easily interpretable	Not easily understandable
Low precision	High precision

A ML model can be interpreted in a variety of ways using Explainable AI methods like agnostisity and scope.

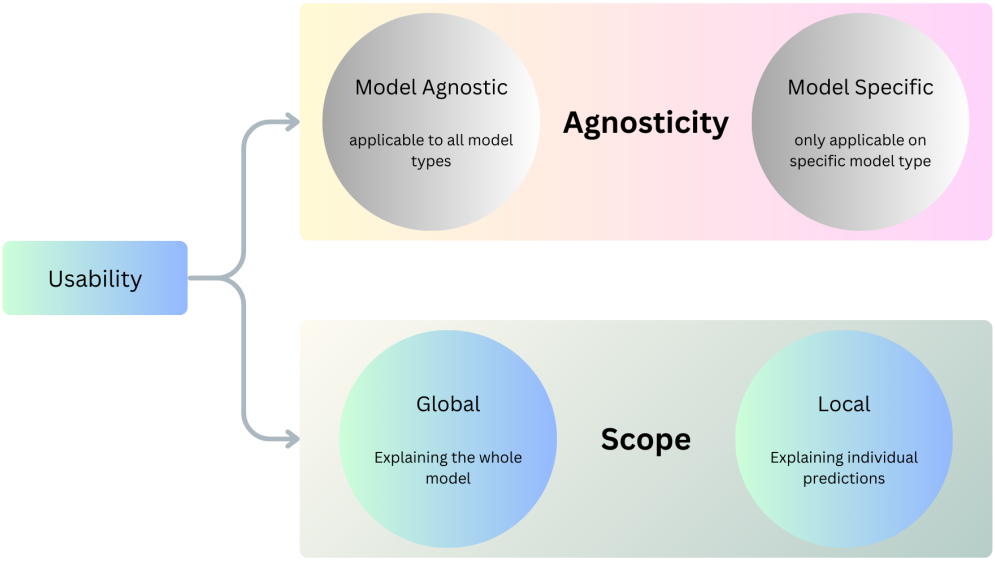


Figure I.3: XAI Usability General Categorization

focusing en the **Global** and **Local** explanation, the first one assists in identifying the predominant features responsible for influencing the model's output, on the other hand the Local explanation Target individual predictions to explain,

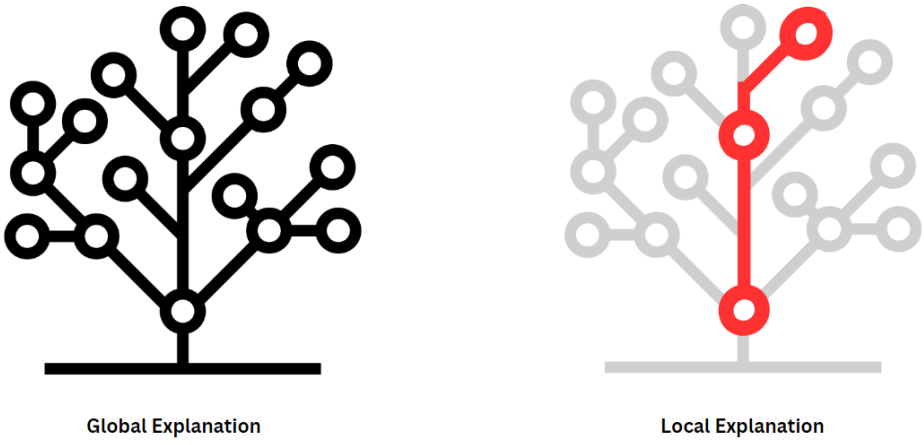


Figure I.4: Global and Local Explanation general idea

A common idea to achieve the local explanation is the perturbation-based strat-

egy, which provides **post-hoc** explanation to the prediction outcome by checking the performance change after perturbing the input features [13], [14].

In the realm of interpretability techniques, methods can be categorized into two broad stages:

1. **Post-hoc method :** Post-hoc methods approximate complex black-box ML models by generating simpler surrogate models [15].These surrogate models allow human examiners to grasp and appreciate the internal mechanisms of black-box models.

2. **Ant-hoc method :** also named "inherent" or "intrinsic", built into the model, from the start, ensuring that the model is inherently interpretable. Ante-hoc methods typically include models like decision trees, linear regression, and rule-based models, which are naturally interpretable. These models, often referred to as "glass box" models, provide clarity on how decisions are made due to their straightforward structure and the direct relationship between inputs and outputs [16].

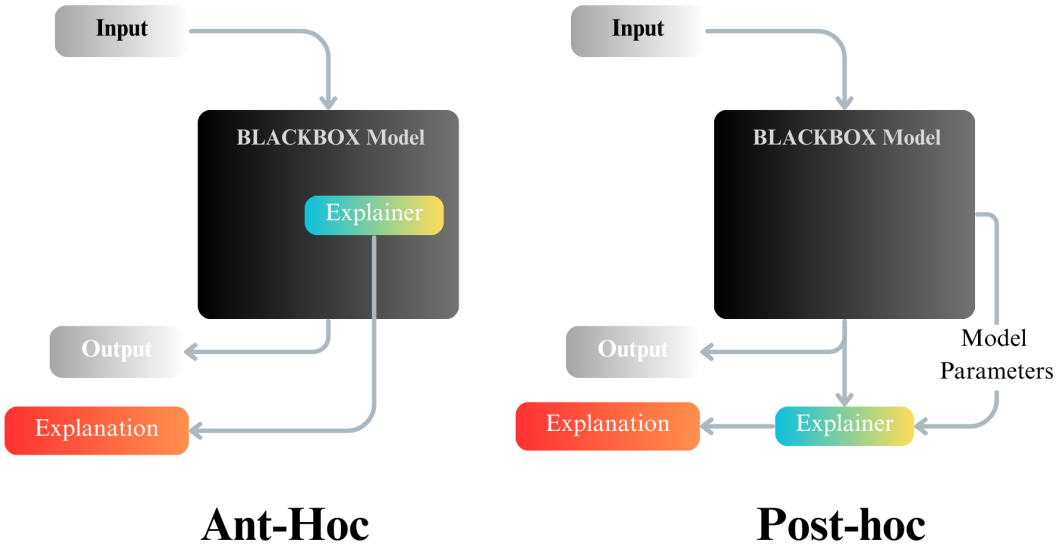


Figure I.5: Difference between Ant-hoc/Post-hoc stages

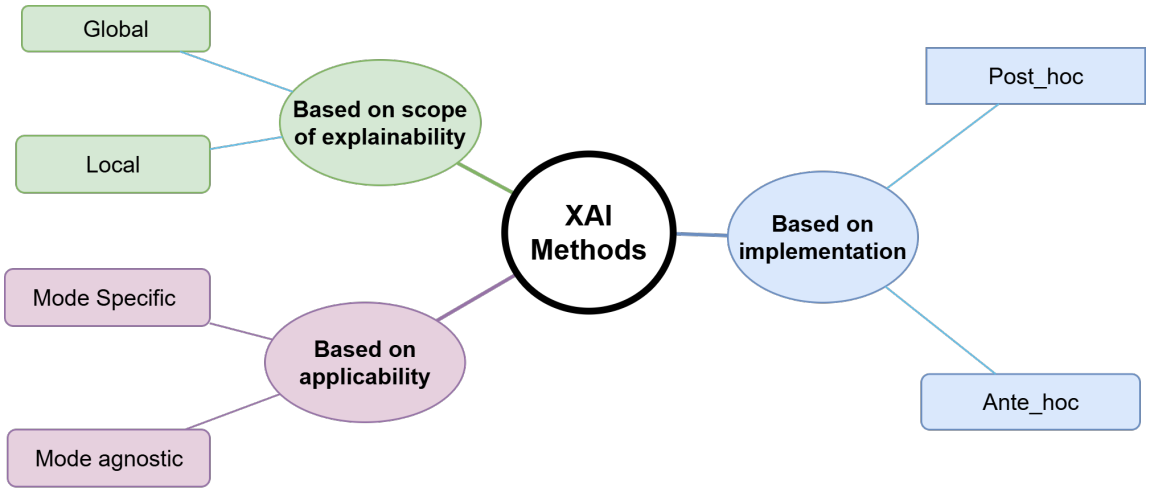


Figure I.6: Differents XAI methods

Several well-known XAI methods illustrate the diverse strategies used to interpret ML model outputs:

1. **Method SHAP (SHapley Additive exPlanations) :** The SHAP approach, a game-theoretic technique, provides an explanation for the output of ML models by distributing the prediction value fairly across multiple inputs. Known for its reliability, it's one of the most popular approaches, blending local interpretations with effective credit allocability to ensure an even distribution of estimates. Only symmetry, dummy, and additivity are the three properties that Shapley values can satisfy among the available solutions[17].

2. **Method LIME (Local Interpretable Model-agnostic Explanations) :** LIME is used to construct simple "surrogate" models that are built on selected data points to understand how a more complex model can make predictions. To train a simpler model, it creates another dataset that is centered on the data point of interest by itself [13]. Any black-box model and a variety of data formats can be utilized with this flexible approach. Despite not uncovering the internal processes of the original model, domain knowledge provides validation and confidence-building capabilities.

I.2.5 Intrusion Detection System

intrusions are defined as efforts made to compromise the confidentiality, integrity, or availability of a computer or network, or to bypass the security measures in place for them. So an IDS is a system (software or hardware) designed to monitor network traffic and promptly notify of any suspicious or malicious activities that are detected [18]. Hence, it is crucial to develop a reliable IDS to counter various types of attacks effectively. IDS serves as a primary defense mechanism for network security [19]. To achieve this, IDS employs various techniques to identify anomalies. IDS can be categorized based on two primary factors : the location of detection (network or host) and the methodof detection employed (signature or anomaly-based) [20].

IDS classification by location of detection

There are two main types of IDS by location of detection : Host Intrusion Detection System (HIDS) and Network Intrusion Detection System (NIDS) [21].

a. **Host-based intrusion detection system :**

   HIDS keeps a log of system events to detect any unusual activities and continuously updates itself with information about device artifacts, operations, and memory regions [22]. It's important to note that a host-based IDS alone is not considered an optimal solution. It has significant drawbacks, such as consuming a high amount of system resources, which negatively impacts the performance of the host. Additionally, certain attacks may go unnoticed unless they successfully breach the host's defenses.

b. **Network-based intrusion detection system :**

   NIDS is strategically positioned at specific locations within the network to analyze traffic from all connected networks [23]. It examines all the network

traffic passing through its subnet and compares it with an anomalies library to identify potential intrusions. NIDS performs this monitoring process discreetly, without raising suspicion.

IDS classification by the method of detection

There are two primary approaches used in IDS : signature-based detection and anomaly-based detection [24].

a. **Signature-based detection :**

Signature-based intrusion detection relies on pattern recognition techniques to identify attacks, often referred to as knowledge-based detection. It maintains a repository of patterns from previous attacks and employs matching algorithms to compare these known patterns with new data. If a new pattern closely resembles a known signature, a warning signal is triggered to indicate a potential intrusion. The main objective is to utilize a library of known attack signatures to identify intrusions [21]. However, signature-based intrusion detection encounters a significant challenge : if an attacker employs a new attack that is not present in the signature library, the method fails to detect the attack. Such attacks are commonly referred to as zero-day attacks [25].

b. **Anomaly-based detection :**

Anomaly-based intrusion detection serves as a solution to the challenges faced by signature-based intrusion detection. It detects intrusions by analyzing user behavior. A baseline or normal model of the system's behavior is established using statistical and other techniques. Any deviation or difference between the actual behavior and the predicted behavior is flagged as an anomaly, potentially indicating an intrusion [21]. However, anomaly-based intrusion detection has its -own limitations. For instance, it struggles to identify attacks within

encrypted packets, leaving a vulnerability. Additionally, creating an accurate normal model for large-scale dynamic data is highly challenging, often resulting in false alarms or incorrect identification of anomalies [26].

IDS classification by time aspect

In considering the temporal aspects of IDSs, it is necessary to distinguish between two main groups: real-time (on-line) IDSs and off-line IDSs [27].

a. **Real-time (on-line) IDSs:**

attempt to detect intrusions in real-time or near real-time They operate on continuous data streams from information sources and analyze the data while the sessions are in progress. Real-time IDSs should raise an alarm as soon as an attack is detected, so that action that affects the progress of the detected attack can be taken.

b. **Off-line IDSs:**

perform post-analysis of audit data. This method of audit data analysis is common among security analysts who often examine network behavior, as well as behavior of different attackers, in an off-line mode. Many early host-based IDSs used this timing scheme, since they used operating system audit trails that were recorded as files.

IDS classification by architecture

There are two principal architectures that are used in IDSs, namely centralized and distributed IDSs [27].

a. **Centralized IDS:**

Most IDSs employ centralized architecture and detect intrusions that occur in a single monitored system.

b. **Distributed IDS:**

there is a recent increasing trend towards distributed and coordinated attacks, where multiple machines are involved, either as attackers (e.g. distributed denial-of-service) or as victims (e.g. large volume worms). Analysis that uses data from a single site and that is often employed by many existing intrusion detection schemes is often unable to detect such attacks. To effectively combat them, there is a need for distributed IDS and cooperation among security analysts across multiple network sites. Unlike a centralized IDS, where the analysis of data is performed on a fixed number of locations (independent of how many hosts are being monitored), in a distributed IDS the analysis of data is performed on a number of locations that is proportional to the number of hosts that are being monitored.

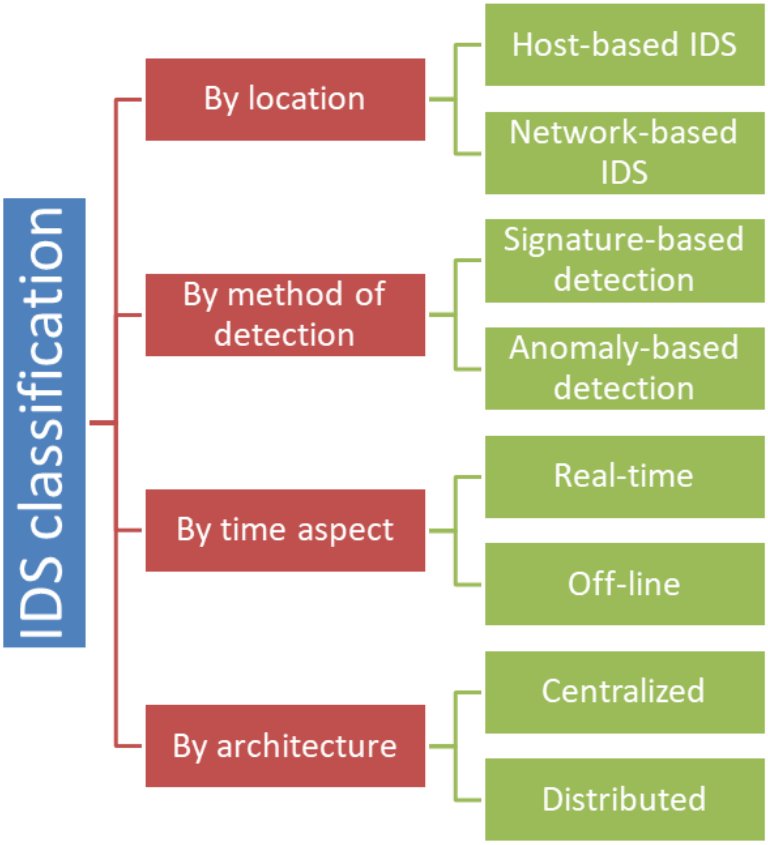


Figure I.7: IDS classification

I.3 Related Work

In the current time, there is still not much work in the field of applying XAI methods to IDS. Two of the most used methods are LIME and SHAP, which are already defined in Chapter 01. In the work [28], a framework is proposed to provide an explanation for IDS using SHapley Additive exPlanations (SHAP), counting on both local and global scopes with the purpose of improving the explanation. This offers multiple advantages such as high transparency. Additionally, the framework compares two classifiers, one-vs-all and multiclass, facilitating adjustments to the IDS structure based on differences in interpretation between these classifiers.

However, using an old dataset like NSL-KDD may pose several inconveniences.

It lacks coverage of new cyber-attacks, potentially missing emerging threat patterns that have evolved since its creation. Moreover, the dataset may not incorporate recent advancements in intrusion techniques and defenses, potentially leading to outdated detection models. also the data set contains only NIDS traffic, Training exclusively on an NIDS traffic data further limits its capability. While proficient in detecting network-level anomalies such as port scans, DoS attacks, and certain exploits, it may struggle with anomalies observable only at the host level, like unauthorized access attempts, file system manipulations, or unusual process behaviors. Additionally, sophisticated attacks often involve both network maneuvers and subsequent actions on compromised hosts. Without exposure to both NIDS and HIDS data, the IDS might miss the complete sequence or context of an attack, leading to incomplete detection and compromised security efficacy.

While in the work [29], they plan to design a framework specifically for evaluating XAI methods in the context of network intrusion detection, distinguishing itself from prior works. This is noted considering that the complexity of the evaluation process can increase while evaluating a diverse range of AI models. The usage of SHAP in this work provides both global and local explanations. However, despite their great work, a significant drawback is that the three datasets used show that the work was done only on network traffic IDS.

The work [30] had an important advantage, the models exhibited high accuracy, reaching 99%. This suggests that the models are effective at distinguishing normal instances from attack ones, which is crucial for reliable intrusion detection. The objective is to apply XAI techniques to enhance interpretability in the models, thus achieving both high accuracy and interpretability. However, the same drawback of the work [29] appears in the dataset used shows that the work was done only on network traffic IDS, also they did not use local scope, This may frequently result inability to provide an explanation for specific decisions, which make it hard for experts to understand individual predictions.

In 2022, Swetha Hariharan et al. [31] conducted a study to investigate different

XAI methods for IDS, focusing on both global and local explanation approaches. The study compared local agnostic methods such as SHAP, LIME, and Contextual Importance and Utility (CUI) with global agnostic methods such as SHAP and Permutation Importance (PI) to assess their ability to explain IDS model behavior. ML models such as Random Forest (RF), XGBoost, and LightGBM (LGBoost) were used and tested on real data from the Kaggle IDS and NSL-KDD datasets, ensuring practical relevance to cybersecurity.

This study has several advantages, such as providing a comprehensive comparison of explanation methods, offering a range of insights into the behavior of IDS models, and showing which techniques provide the most effective explanations. By leveraging a range of explanatory methods, such as SHAP and LIME, the study allows for a broader and more robust analysis of IDS models using real-world datasets like Kaggle IDS and NSL-KDD. Despite these advantages, the study's lack of experimentation, as the solution is tested on only one dataset, may undermine the robustness and generalizability of its findings.

In 2023, Harry Chandra Tanuwidjaja et al. [32] introduced the Hybrid Explainable Intrusion Detection System (X-IDS) framework, a significant development in the analysis of the Ton-IOT IDS dataset, a Windows-based dataset released by the University of New South Wales in 2021. The X-IDS framework uses different explanatory methods to improve the understanding of various types of cyber-attacks. By utilizing variable importance plots, which highlight key model features, individual value plots, which visualize specific data points, and partial dependence plots, which show how individual features affect predictions in general, this framework provides a comprehensive perspective on cybersecurity threats.

The framework's ability to leverage a range of explanatory methods, such as SHAP and LIME, allows for a broader and more robust analysis of IDS models. It helps identify discrepancies between explanatory techniques, providing a more rounded approach to explainability. However, this study has some drawbacks, including relying solely on NIDS traffic and training models on outdated datasets.

In 2023, Mohammed M. Alani et al. [33] presented an explicable ensemble-based approach for detecting cyber-attacks on the Internet of Medical Things (IoMT). The study used a combination of multiple classifiers through a soft voting ensemble method to improve accuracy. The classifiers included Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVC), and Extreme Gradient Boosting (XGB). The research utilized the WUSTL-EHMS-2020 dataset and employed SHAP (SHapley Additive exPlanations) to provide insight into feature contributions.

The study has many advantages, such as an effective ensemble approach, where the soft voting ensemble method combined predictions from multiple classifiers, resulting in higher accuracy and reduced false positives and negatives. Additionally, preprocessing steps ensured a balanced representation of normal and attack traffic, reducing bias and improving model robustness.

Despite these advantages, the study has some weaknesses. The results are specific to the WUSTL-EHMS-2020 dataset, which may limit their generalizability to other datasets or environments. Furthermore, the study exclusively utilized the SHAP method, overlooking the potential benefits of integrating LIME for a more comprehensive approach to explainability. Additionally, the test was conducted on only one dataset.

In 2024, Diogo Gaspar et al. [34] sought to improve the interpretability of a black-box intrusion detection system (IDS) for Internet of Things (IoT) devices. Using the ADFA-LD dataset, they applied two XAI methods, LIME and SHAP, to identify key system calls that influenced the predictions of the IDS model. To validate these explanations, they performed a perturbation analysis where they changed or removed the top 10 system calls identified by LIME and SHAP and observed how these changes affected the model's output. They also conducted a survey analysis to measure the perceived clarity and trustworthiness of the XAI explanations from the participants' perspective.

This study covers several benefits, such as demonstrating the value of LIME and SHAP in improving the transparency of IDS models and providing useful insights into

the most influential features. It improved transparency by identifying the key system calls that influenced the predictions of the IDS model. It also facilitated model refinement by allowing targeted improvements to be made, focusing on correcting specific system calls that could lead to misclassification.

Despite these advantages, relying solely on HIDS traffic restricts detection to host-specific threats, such as internal infiltrations and user activity logs, potentially overlooking broader network-based attacks identifiable by NIDS. Additionally, the test was conducted on outdated datasets.

I.4 Conclusion

This overview begins by defining AI and ML, including its types and role in enhancing AI capabilities. It explores DL, a subset of ML with multi-layered neural networks. The analysis highlights the importance of Intrusion Detection Systems IDS in network security, categorizing them into NIDS and HIDS. Explainable AI (XAI) techniques, such as post-hoc explanations and interpretable models, are discussed to improve understanding and trust in AI decisions.

Reviewing XAI for IDS shows its growing importance in cybersecurity. XAI methods like LIME and SHAP help validate and refine models by making black-box models transparent, aiding cybersecurity professionals in understanding and strengthening strategies. However, challenges exist. Many IDS datasets are outdated or specific to HIDS or NIDS, limiting generalizability and potentially misrepresenting modern threats. Limited experimentation in some studies raises concerns about the robustness of XAI methods.

Ref	Scop	methods	AI Models	Datasets	Advantages	Drawbacks	Usability	Stage
[28]	local global	SHAP	One-vs-All Classifier Multiclass Classifier	NSL-KDD	Utilizing both one-vs-all and multiclass classifiers	- Old dataset used - NIDS traffic only - Lack of experimentation	agnostic	post-hoc explanation
[29]	local global	LIME SHAP	LGBM, DNN, MLP, CNN, RF, SVM, ADA, KNN	NSL-KDD SIMARGL2021 CIC-IDS2017	- Use new datasets - Use multiple datasets	- NIDS traffic only	agnostic	post-hoc explanation
[30]	global	SHAP	LightGBM CNN	CICIDS 2017	High Model Accuracy	- No use for the local scope - NIDS traffic only	agnostic	post-hoc explanation
[31]	local global	SHAP LIME	RF XGBoost LightGBM	Ton-IOT IDS	- Comprehensive Analysis - use different techniques - Detailed Comparison of SHAP and LIME	- Lack of experimentation	agnostic	post-hoc explanation
[32]	local global	SHAP LIME CUI PI	RF	NSL-KDD Kaggle	- Comparison of Methods Global and Local Scope - Tests on Multiple Models of ML - Real-World Data	- Use old dataset analyse - NIDS traffic only	agnostic	post-hoc explanation
[33]	local global	SHAP	RF, DT, SVC, XGB	NSL-KDD WUSTL-EHMS-2020	- Use a combination of multiple classifiers - Effective Ensemble Approach - Balanced Dataset	- Dataset Specificity - use only SHAP - Lack of experimentation	agnostic	post-hoc explanation
[34]	local global	LIME SHAP	MLPClassifier	ADFA-LD	- Improved Transparency - Model Refinement use SHAP and LIME	- Use only HIDS - Old dataset	agnostic	post-hoc explanation

Table I.1: Summary Table of All Previous Articles:

Chapter II

METHODOLOGY

II.1 Introduction

In this chapter, the drawbacks identified in the previous research will be addressed, and the advantages will be leveraged to enhance the proposed model.

One question often arises when discussing the integration of XAI: "Why use XAI if we already have AI models with high accuracy?" This is a crucial question. Although current AI models can provide accurate results, they often lack interpretability. Users frequently find themselves asking the model, "Why did you do that?" or "How did you come up with that result?" or "who do i correct error?" These questions highlight the significant weakness of traditional AI models their lack of transparency.

The figure I.2 illustrates the importance of using XAI to explain and interpret the decisions made by black-box AI models. Interpretability is the main reason for moving to XAI. Being able to understand "why" a model makes certain decisions makes the model clearer and more transparent. This transparency serves multiple purposes, the most important being increased trust in the explained model. By integrating XAI methods, users can gain a better understanding of the model's decision-making process, leading to more informed and confident use of AI systems

II.2 Proposed Architecture

The proposed IDS architecture integrates internal networks and a Demilitarized Zone (DMZ) hosting critical services, including a web server, email server, and a powerful server dedicated to advanced AI solutions utilizing models such as XGBoost and ANN and XAI methods such as LIME and SHAP. This integration enhances the security of the infrastructure. A centralized logging server has been implemented on a powerful server.

The strategy for defense includes a hardware-based NIDS for real-time analysis of network traffic, supplemented by software-based HIDS deployed across all workstations and servers. Each HIDS continuously forwards alerts and logs to the central-

ized logging server. Upon detecting malicious activity, whether identified by HIDS or NIDS, immediate notifications are relayed to the administrator situated at the management station. The administrator then evaluates the threat, decides whether to allow or block the traffic by adjusting firewall configurations, and finally utilizes collected data to train AI models housed on the powerful server. These AI models are trained on several types of attacks, including Reconnaissance, Backdoor, DoS, Exploits, Worms, Shellcods, Analyses, Reconnaissance, Gneric and Infiltration from inside the network. Subsequently, both local and global XAI methods are employed on AI models to interpret and elucidate the decisions made by the IDS. This comprehensive approach ensures a thorough understanding of security events and facilitates proactive measures against potential threats. The following figure illustrates all the previous details.

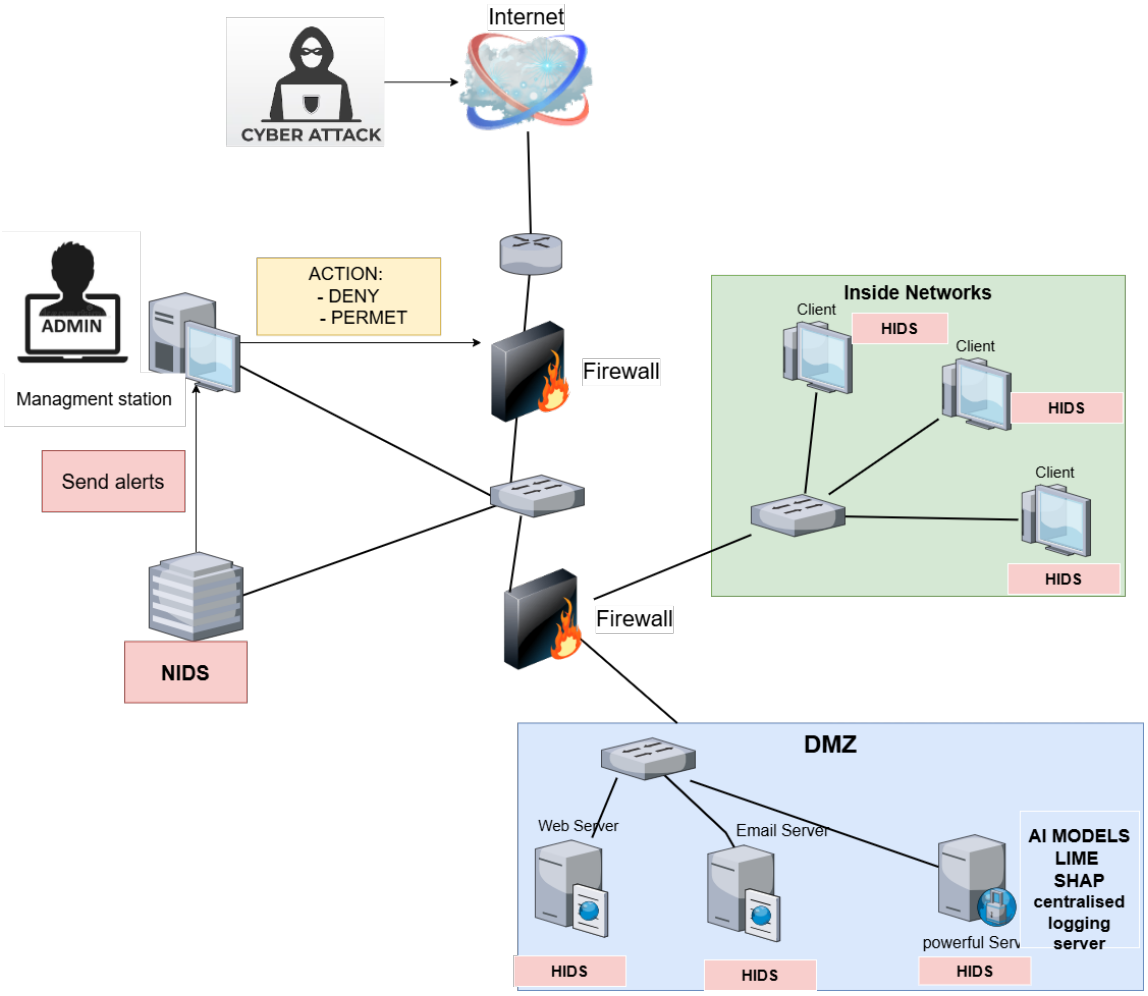


Figure II.1: Proposed architecture for IDS

II.3 Methodology

This section presents a summary of the proposed methodology for XAI methods. The datasets were selected as they are readily available on the internet. Figure II.2 represents the workflow for achieving the objectives of this study. This diagram illustrates the detailed steps for using XAI methods to interpret the predictions of a black-box AI model, as outlined below:

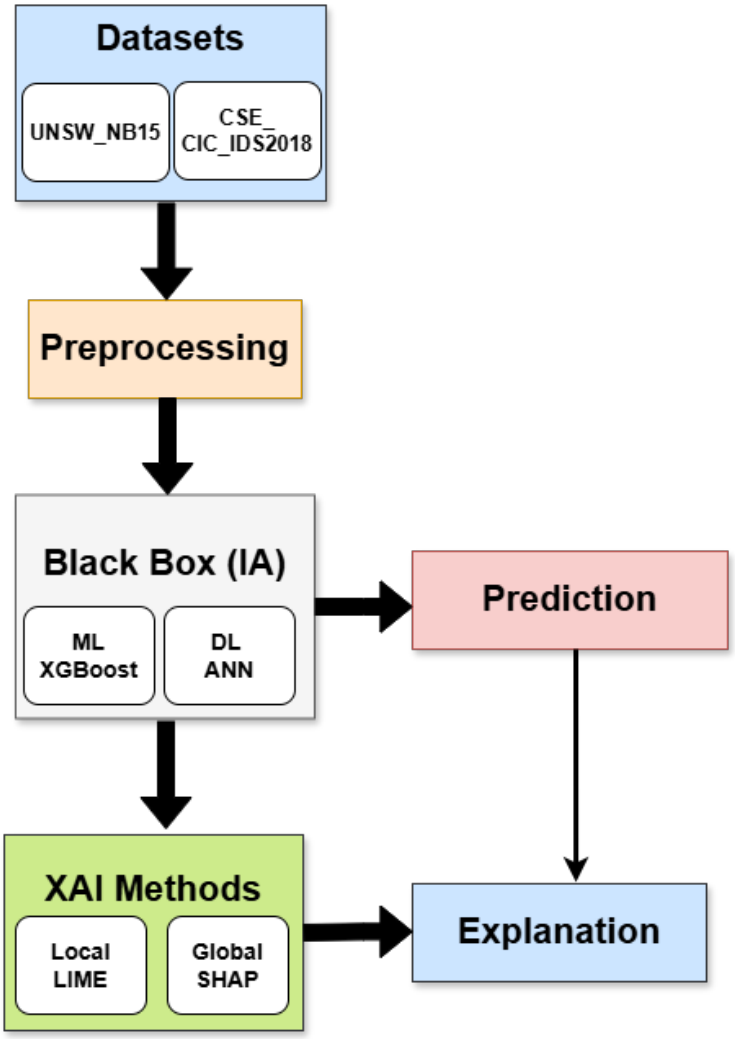


Figure II.2: XAI Model

II.3.1 Data Selection

The initial step in the XAI process is data collection, which involves gathering comprehensive datasets to train the AI models (XGBoost for ML and ANN for DL). For IDS, typical datasets include network traffic datasets (UNSW-NB15) and combined NIDS and HIDS traffic datasets (CSE-CIC-IDS2018), encompassing both benign and malicious activities. These datasets provide labeled examples of various cyber attacks and normal traffic, ensuring the AI models can identify patterns and accurately distinguish between benign and malicious traffic.

II.3.2 Pre-processing:

In this step, the raw data from the dataset is cleaned and transformed into a suitable format for the AI model. The purpose of this step is to make the data ready for the AI model, this include the following tasks.

1. **Data cleaning :** starting by identifying and addressing missing values, outliers, and noise within both datasets. This meticulous process was essential to enhance data integrity and prevent distortions in analysis.

2. **Normalization :** Continuous features in both datasets were normalized or standardized to establish a uniform scale. This standardization mitigated the impact of varying scales on subsequent analysis and modeling.

3. **Label encoding :** While some labels in both datasets were already in the valid format of 0 and 1, certain samples contained invalid formats. Consequently, a comprehensive label encoding step was executed to ensure consistent formatting across all labels. This transformation facilitated seamless integration with machine learning algorithms.

4. **Dataset splitting :** To facilitate rigorous model training and validation, the preprocessed datasets were divided into training and validation subsets. This

division ensured that models could be effectively trained on one subset and then evaluated on the other, promoting robust performance assessment.

II.3.3 BLACK BOX (AI):

The study employed two distinct AI models, each of which was optimised to leverage its respective strengths for training and prediction purposes. The first model is the ML model, XGBoost.

1. **XGBoost** [35] (Extreme Gradient Boosting), which is renowned for having a high efficient and scalable implementation of the gradient boosting framework, which builds models in a stage-wise fashion. This approach optimizes a differentiable loss function by iteratively adding weak learners, typically decision trees, to minimize the loss, The loss function in XGBoost can be expressed as:

$$L(\theta) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k) \tag{II.1}$$

where l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. The term $\Omega(f_k)$ represents the regularization term for the k-th weak learner f_k. One of the key features of XGBoost is its inclusion of regularization terms, specifically L1 (Lasso) and L2 (Ridge) regularization, to prevent overfitting, which is not traditionally part of the gradient boosting framework [36]. also, XGBoost includes a built-in cross-validation mechanism at each iteration of the boosting process, ensuring that model tuning and selection are performed efficiently. The second model is the DL model, which is ANN.

2. **ANN** [37] consist of layers of interconnected nodes (neurons). These layers are typically categorized as the input layer, which receives the initial data,

hidden layers, which are intermediate layers where computations and feature transformations occur, and the output layer, which produces the final output. Each neuron in an ANN processes input data and passes the result to the next layer. The processing involves weights, where each input to a neuron has an associated weight indicating the importance of that input, and a bias, which is an additional parameter in the neuron that helps adjust the output. The weighted sum (linear combination) of inputs can be expressed as: The weighted sum (linear combination) of inputs can be expressed as:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{II.2}$$

where:

- z is the weighted sum.
- x_i are the input features.
- w_i are the weights associated with each input feature.
- b is the bias term.
- n is the number of input features.

Common activation functions [38] include the sigmoid function, which outputs values between 0 and 1, the ReLU (Rectified Linear Unit) function, which outputs the input directly if it is positive, otherwise it outputs zero, and the tanh function, which outputs values between -1 and 1.

II.3.4 Prediction:

The trained AI model employs techniques from both ML and DL to generate predictions on new input data by leveraging the intricate patterns and relationships it

has assimilated during the training phase. This process entails the model applying its internalised knowledge to meticulously analyse and interpret the new data, thus producing outputs or predictions that are deeply informed by its prior learning. The objective is to utilise the model's capabilities to deliver accurate and reliable predictions on unseen data, thereby demonstrating its capacity to generalise effectively beyond the training set.

II.3.5 XAI Methods:

Explainable AI methods enhance transparency and trust in black-box AI models by providing local and global explanations of their predictions.

1. **LIME :** a method that focuses on individual predictions. It approximates the black-box model with an interpretable one in order to explain the rationale behind a specific instance.

   (a) **LIME main characteristics [13]:**

      i. **Local Fidelity:** LIME focuses on making the explanation accurate around the prediction being explained.

      ii. **Model-Agnostic:** LIME can be applied to any ML model, regardless of its complexity or structure.

      iii. **Simplicity:** By using simple interpretable models to approximate the complex model, LIME ensures that the explanations are easy to understand.

   (b) **LIME working Steps [39]:**

      i. **Step 1: Select an Instance:** Choose the instance (data point) for which you want to generate an explanation. Let's denote this instance as x.

ii. **Step 2: Generate Perturbed Samples:** Create a new dataset by perturbing x. Perturbations involve slightly modifying the feature values of x. This generates a set of new samples around x. The perturbed samples x_i' can be generated as:

$$x_i' = x_i + \epsilon_i \tag{II.3}$$

where ϵ_i is small random noise, typically drawn from a normal distribution.

iii. **Step 3: Predict Perturbed Samples:** Use the black-box model to predict the outcomes for these perturbed samples. This provides a set of predictions corresponding to the perturbed dataset.

iv. **Step 4: Weight the Perturbed Samples:** The perturbed samples are weighted based on their proximity to the original instance x. A common weighting function used is the exponential kernel, which assigns higher weights to samples closer to x. The weight w for a perturbed sample x' is calculated as:

$$w(x, x') = \exp\left(-\frac{D(x, x')^2}{\sigma^2}\right) \tag{II.4}$$

where $D(x, x')$ is the distance between x and x', and σ is a kernel width parameter controlling the rate of decay of the weights.

v. **Step 5: Train an Interpretable Model:** Fit a simple interpretable model (e.g., a linear regression or decision tree) to the weighted perturbed dataset. The goal is to approximate the complex model locally around x.

vi. **Step 6: Generate the Explanation:** Extract the parameters

(e.g., coefficients) of the interpretable model to understand the contributions of each feature to the prediction for x. These parameters form the explanation.

2. **SHAP [40]:** offers a comprehensive understanding of the model's overall behaviour by assigning importance values to each feature based on game theory. This elucidates the significance of features across all predictions.

   (a) **SHAP main characteristics [40]:**

      i. **Consistency :** SHAP values guarantee that if a model changes in a way that increases a feature's contribution to the prediction, the Shapley value for that feature will not decrease.

      ii. **Model-Agnostic:** SHAP can be used with any ML model, making it very versatile.

   (b) **SHAP working Steps:** how SHAP works can be devided in 3 steps :

      i. **Step 1: Define the cooperative game :** Each feature is a player in the game. The gain from the coalition of features is the prediction made by the model[40].

      ii. **Step 2: Calculate Shapley Values [41]:** For each feature, the Shapley value is calculated as the average marginal contribution of the feature across all possible subsets of features. Mathematically, for a feature i:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right] \qquad \text{(II.5)}$$

      where N is the set of all features, S is a subset of features not including feature i, $f(S)$ is the model prediction using only the features in subset S, and $f(S \cup \{i\})$ is the model prediction using the

features in subset S plus feature i.

iii. **Step 3: Interpret the Values :** The calculated Shapley values represent the contribution of each feature to the prediction for a given instance. These values can be aggregated across instances to understand the overall impact of each feature on the model's predictions.

II.3.6 Explanation:

The final stage is explanation, XAI methods involves generating clear explanations for the AI model's predictions. These explanations help cybersecurity experts understand the model's decision-making process, fostering trust and validation of its outputs. This transparency is crucial for identifying biases, improving accuracy, and ensuring compliance. By providing detailed insights, XAI enhances collaboration between human expertise and AI technology, boosting the reliability and acceptance of AI systems, especially in distinguishing between attack and benign traffic.

II.4 Conclusion

This chapter presents a comprehensive methodology for developing, evaluating, and interpreting AI models by using two different types of datasets. The objective is to create models that are not only robust and accurate but also transparent and interpretable. To this end, ML and DL techniques are integrated with XAI methods. The use of local explanation techniques, such as LIME, and global explanation methods, such as SHAP, helps to demystify the black-box nature of AI models, thereby enhancing their interpretability. The following chapter will provide a detailed account of the preceding steps, accompanied by a comprehensive analysis of the results of our experiments.

Chapter III

IMPLEMENTATION AND RESULTS

III.1 Introduction

In this chapter, the methodology proposed in Chapter II is implemented by applying ANN and XGBoost models to the UNSW_NB15 and CIC-IDS-2018 datasets, followed by model explanations using SHAP and LIME XAI methods. The preprocessing steps are outlined, including data cleaning and normalization, along with a detailed explanation of the performance metrics used: accuracy, precision, recall, F1 score, and confusion matrix. Results from SHAP and LIME interpretations of model predictions are then presented, enhancing transparency and explainability. Finally, the findings and their implications for improving intrusion detection systems are discussed, offering insights for further research and development.

III.2 Development Environment

In order to implement and evaluate our proposed system, by utilizing a range of powerful tools. Google Colab was employed for its cloud-based computational resources and collaborative features, enabling us to leverage scalable computing power for model training and experimentation [42]. Python [43] served as the primary programming language, with TensorFlow and Keras used extensively for building and training deep learning models, ensuring robust performance and flexibility in model development [44] [45]. In the local environment, Anaconda provided a comprehensive distribution for managing packages and dependencies, ensuring consistency across development setups [46]. Jupyter Notebooks were instrumental in facilitating interactive coding and data analysis, enhancing our ability to explore data insights and iterate on model designs effectively [47].

III.3 Dataset

Nowdays Searching for a datasets is a real challenge, the dataset chosed should contain both NIDS and HIDS traffic to recover much more percetnage of security, and also it's very important to take in consideration the release date of that dataset therefore these two datasets will include both advantages :

III.3.1 CSE-CIC-IDS2018 Dataset :

The CSE-CIC-IDS2018 dataset is the product of a collaborative project between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC). It was created using a systematic approach based on the notion of profiles to generate comprehensive cybersecurity data. The dataset provides detailed descriptions of various intrusions, along with abstract distribution models for applications, protocols, and lower-level network entities. It encompasses seven different attack scenarios: Brute-force, Botnet, DoS, DDoS, Web attacks, and infiltration of the network from inside [48]. The CSE-CIC-IDS2018 dataset comprises 1,482,109 rows and 80 columns (features), spanning a total size of 10 GB. Despite its vast scale, I utilize a more manageable subset of around 500 MB from this dataset.

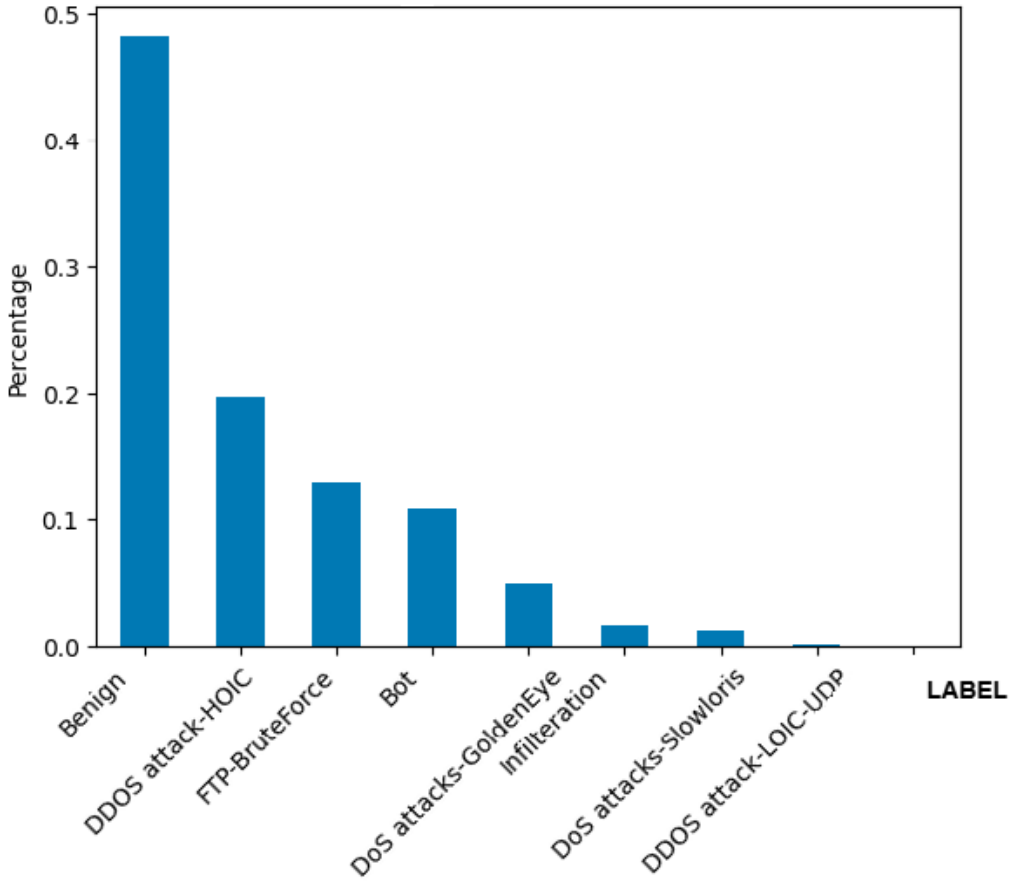


Figure III.1: Diagram of attacks types (CSE-CIC-IDS2018)

III.3.2 UNSW-NB15 Dataset :

The UNSW-NB15 dataset is a comprehensive collection of network traffic data designed for research in network intrusion detection. Developed by the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) in 2015, ths dataset was updated, the last updated version was released in 2021 it contains 82332 rows and 45 columns. the dataset aims to provide a realistic and challenging benchmark for evaluating IDS [49]. It includes a variety of contemporary attack types and normal network behaviors, representing modern network traffic more accurately than prior datasets such as KDD99.

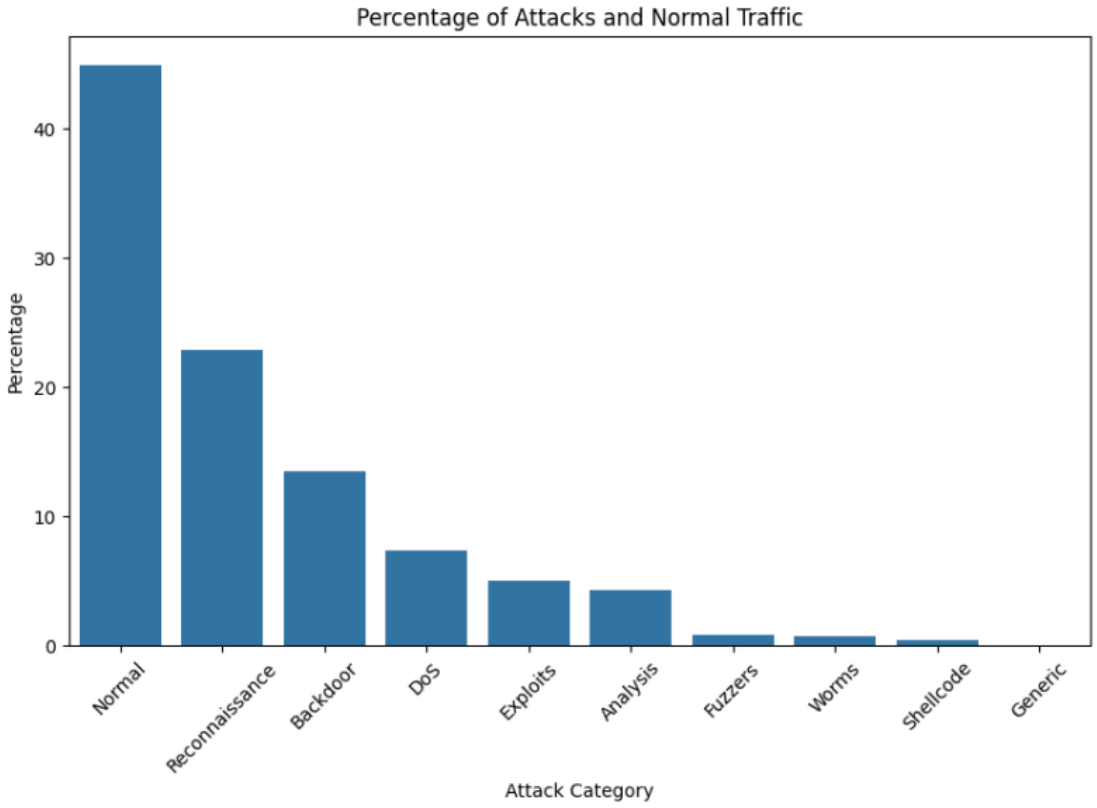


Figure III.2: Diagram of attacks types(UNSW_NB15)

III.3.3 Attacks types :

The two datasets includes several different types of modern attacks such as (Denial of Service DoS, Exploits, Fuzzers, Generic, Reconnaissance, ShellCode, Worms, Infiltration, DDos and Botnet) alongside normal traffic.

1. **Reconnaissance:** A reconnaissance cyberattack is an initial phase where attackers gather information about a target system or network to identify vulnerabilities. This includes activities like network scanning and social engineering. The data collected helps plan more targeted and effective future attacks [49].

2. **Backdoor:** A backdoor attack involves inserting a hidden malicious mechanism into a system, allowing unauthorized access or control, often bypassing normal authentication procedures [50].

3. **DoS:** A Denial of Service (DoS) attack is an attempt to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the internet [49].

4. **DDoS:** A distributed denial-of-service DDoS attack is initiated by a vast array of malware-infected host machines controlled by the attacker. These are referred to as "denial of service" attacks because the victim site is unable to provide service to those who want to access it. [51]

5. **Botnet:** A botnet refers to a group of computers infected with malware and controlled by a malicious actor. The term botnet is a combination of robot and network, each infected device is referred to as a bot. Botnets can be designed to perform illegal or malicious tasks, including sending spam, stealing data and DDoS attacks. [52]

6. **Exploits:** An exploit is a piece of code or a sequence of commands that takes advantage of a vulnerability or bug in a software application, operating system, or hardware device to cause unintended or unanticipated behavior, often to gain unauthorized access or elevate privileges [49].

7. **Worms:** A type of malicious software that self-replicates and spreads across computers and networks without needing to attach to a host program. They exploit vulnerabilities in operating systems or applications to propagate, often causing widespread disruption and damage [49].

8. **Shellcods:** Shellcodes are small pieces of code used as payloads in the exploitation of software vulnerabilities. Typically written in assembly language, they are executed by an attacker to gain control of a compromised system [49].

9. **Infiltration:** An infiltration cyberattack from inside involves a malicious actor gaining unauthorized access to a system or network from within the

organization. This can be achieved through compromised employee credentials, insider threats, or exploiting internal vulnerabilities [48].

10. **BruteForce:** A brute force attack is a hacking method that uses trial and error to crack passwords, login credentials, and encryption keys. It is a simple tactic for gaining unauthorized access to individual accounts and organizations' systems and networks. The hacker tries multiple usernames and passwords, often using a computer to test a wide range of combinations, until they find the correct login information [53].

III.4 Preprocessing Steps

Effective preprocessing of the network and host datasets is done using the following steps:

1. **Transform Categorical Values into Numerical:** Convert all categorical values into numerical representations using one-hot encoding.
2. **Handling Missing Values:** Identify and address any missing values in the dataset by imputing with mean or median values, or by removing rows or columns with substantial missing data.
3. **Normalize Data:** Standardize the dataset using StandardScaler to ensure all features have a consistent scale, enhancing the model's performance.
4. **Splitting the Data:** Divide the dataset into 80% for training and 20% for testing to train and fine-tune the model.

III.5 Metrics used

In ML, the evaluation of model performance is contingent upon the utilisation of key metrics, including accuracy, recall, precision, F1-score and confusion matrix.

These metrics serve to assess the model's capacity to predict outcomes in binary classification tasks, thereby offering indispensable insights into its efficacy.

1. **Accuracy** the proportion of correctly predicted instances (both true positives and true negatives) among the total number of instances evaluated [54]. Mathematically, accuracy is expressed as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} \tag{III.1}$$

   The number of Correct prediction is referd to TP (True Positive) plus TN (True Negative), and the Total Number of predection is referd to as TP plus TN plus FP (False Positive) plus FN (False Negative).

2. **Recall (sensibility)** It is the proportion of correctly classified positive cases to the total of positive cases that are correctly classified and negative cases that are incorrectly classified [55].

$$\text{Recall} = \frac{TP}{TP + FN} \tag{III.2}$$

3. **Precision** It is the proportion of correctly classified positive cases to the total of the positive cases that are correctly and incorrectly classified [55].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{III.3}$$

4. **F1-score** It establishes the model's accuracy for each class [56].

$$\text{F1-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{III.4}$$

5. **Confusion Matrix** A confusion matrix showed in table III.1 is a table that visualizes the performance of a ML model by summarizing the number of correct

and incorrect predictions made across classes. It compares predicted values with actual values to show the model's accuracy in classification tasks [57].

True Labels	**Predicted**	
	Positive	**Negative**
Positive	TP: When a real Attack is correctly predicted	FP: When a real Attack is wrongly predicted
Negative	FN: When a fake Attack is wrongly predicted	TN: When a fake Attack is correctly predicted

Table III.1: Confusion matrix parameters

III.6 Algorithms Used

This section outlines the algorithms used for training, detailing the steps involved in model training, as well as the functions and parameters utilized in the implementation of the models.

1. **XGBoost (Extreme Gradient Boosting) :** XGBoost, a supervised learning algorithm developed by Chen and Guestrin in 2016, represents an efficient and scalable implementation of the gradient boosting framework. Widely utilized for regression, classification, and ranking tasks, its performance and speed have made it a popular choice in ML competitions and applications [36]. During the application of the XGBoost model on the dataset, several key functions and parameters were utilized, including:

   (a) **Train and Validate XGBoost Model:** initiate XGBoost classifier giving it multi-class logarithmique loss as as value for evaluation metric parameter, setting 42 as the random seed for reproducibility, and 100 the number of trees in the ensemble with 3 as maximum depth for each tree, then train the model with the model fit function.

   (b) **Predict on the Test Set:** Use the model that have been trained previously to make predictions on the test data, using the predict function

and giving it test data.

(c) **Evaluate the Model:** 4 metrics were calculated to evaluate the model Accuracy, Precision, Recall and F1-Score.

2. **ANN (Artificial Neural Networks):** Artificial Neural Networks ANN are computational models inspired by the human brain's structure and function. They are designed to recognize patterns, learn from data, and make decisions, consist of interconnected layers of nodes, or "neurons, " each capable of performing simple computations. [58] in the process of applying the ANN model on the dataset several functions and parameters used for:

   (a) **Creating the model:** Sequential functions to create sequential model and Dense functions to have fully connected layer. For both datasets UNSW_NB and CSE-CIC-IDS2018, three hidden layers were defined, containing 128, 64, and 32 neurons respectively, each using the ReLU activation function. For the output layer, the sigmoid activation function was chosen to perform binary classification.

   (b) **Compiling the Model:** Using the model compile function loaded with the optimizer adam, binary crossentropy for the loss parameter and accuracy used as matrics.

   (c) **Training the Model:** using the model fit function that include trainig data as parameter, the batch size was set for 64 and there were 30 epoch to complete.

   (d) **Evaluating the Model:** computing the metrics including accuracy precesion recall and f1-score using the testing and predicted parameters.

III.7 Results and discussion:

In this subsection, the results of various metrics evaluated on different models using different datasets are presented.

III.7.1 Evaluation of Key Metrics:

The table III.2 presents the results obtained from our experiment, which utilized two different datasets, UNSW-NB15 and CSE-CIC-IDS2018 . by applying two different AI models.

N°	**Model**	**Datasets**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
01	XGBoost	UNSW-NB15	0.97	0.98	0.97	0.97
02	XGBoost	CSE-CIC-IDS2018	0.93	0.94	0.93	0.93
03	ANN	UNSW-NB15	0.96	0.97	0.96	0.96
04	ANN	CSE-CIC-IDS2018	0.95	0.94	0.91	0.92

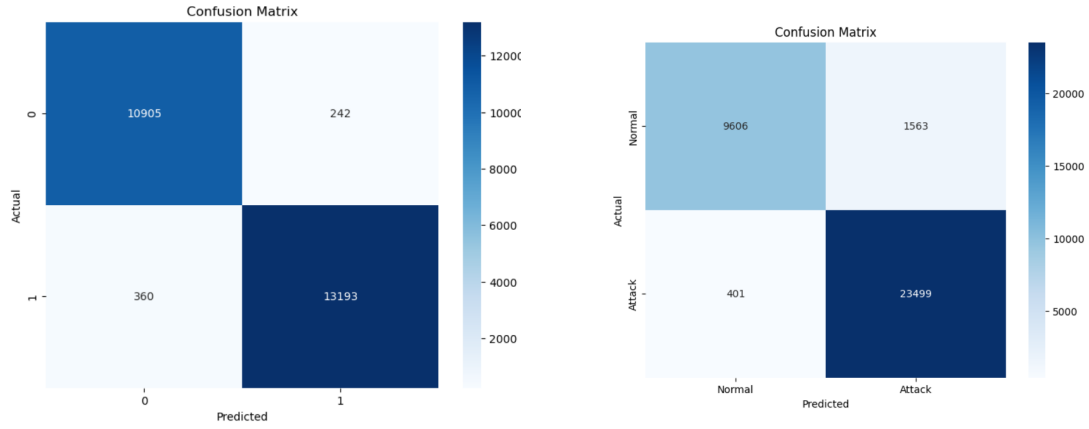
Table III.2: Table of results

the first model is the ML model XGBoost, and the second DL model ANN. The performance of these models was evaluated using various metrics including accuracy, recall, precision, F1-score and confusion matrix. Both models demonstrate high accuracy across the datasets, indicating their reliability in making correct predictions, with XGBoost slightly outperforming ANN on the UNSW-NB15 dataset. Precision, which measures the accuracy of positive predictions, is marginally higher for XGBoost compared to ANN on both datasets, suggesting XGBoost is better at minimizing false positives. Recall, which assesses the ability to correctly identify all positive instances, is similar for both models on the CSE-CIC-IDS2018 dataset.

However, XGBoost has a slightly higher recall on the UNSW-NB15 dataset, indicating better performance in identifying actual positives (malicious traffic). The F1-Score, representing a balance between precision and recall, shows XGBoost has a slight edge over ANN on both datasets, particularly on the UNSW-NB15 dataset, highlighting its better overall balance between precision and recall.

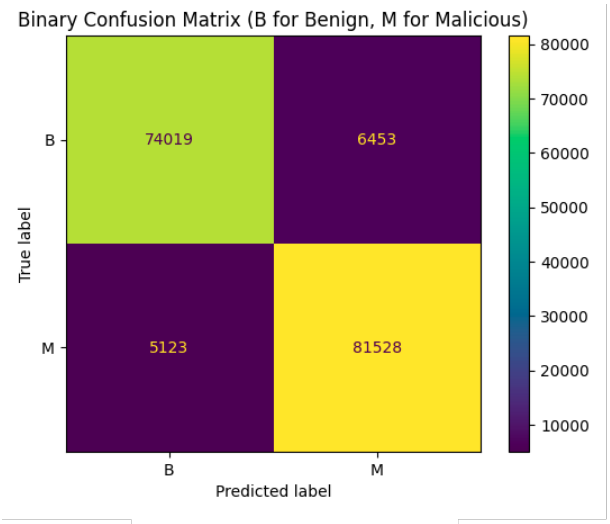
III.7.2 Confusion matrix:

This subsection contains an analysis of the performance of two machine learning models: **XGBoost** and **ANN**, evaluating their ability to correctly classify normal and malicious traffic based on their respective confusion matrices.



((a)) XGBoost ((b)) ANN

Figure III.3: Confusion matrix of models applied on (UNSW_NB15)



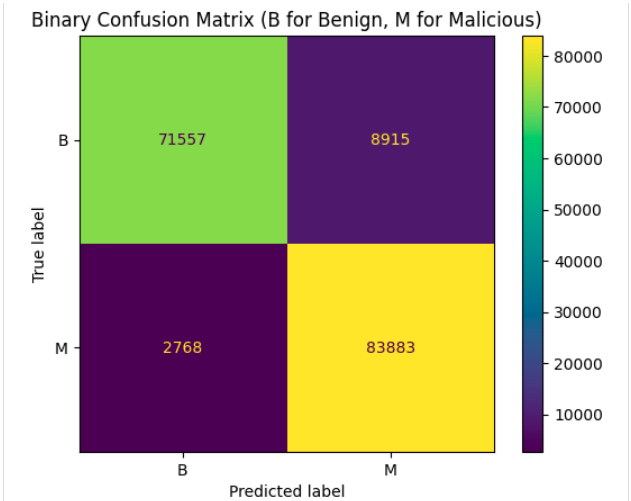


Figure III.4: Confusion matrix XGBoost/ANN (CSE-CIC-IDS2018)

1. **XGBoost Model Performance:**

   The **XGBoost** model as illustrated in Figure III.3 demonstrates varying performance across different datasets. On the **UNSW-NB15** dataset, it accurately identified **10,905** instances of normal traffic, highlighting its proficiency in benign traffic detection. However, it misclassified **360** instances of malicious traffic as normal, indicating potential challenges in distinguishing certain types of attacks.

   Conversely, when applied to the **CSE-CIC-IDS2018** dataset, the **XGBoost** model exhibited robust performance metrics. It correctly classified **81,528** instances of malicious traffic and **74,019** instances of Normal traffic, demonstrating its effectiveness in detecting attacks. Nevertheless, the model encountered **5,123** false positives and **6,453** false negatives, reflecting the dataset's complexity in managing accurate classifications across both Normal and malicious traffic.

1. **ANN Model Performance:**

   Comparatively, the **ANN** model, depicted in Figure III.4, demonstrated consistent performance across both datasets. On the **UNSW-NB15** dataset, it accurately predicted **23,499** out of **25,062** attacks and correctly identified **9,606** instances of normal traffic. This high accuracy underscores the **ANN** model's capability in distinguishing between benign and malicious traffic, making it suitable for reliable traffic classification tasks.

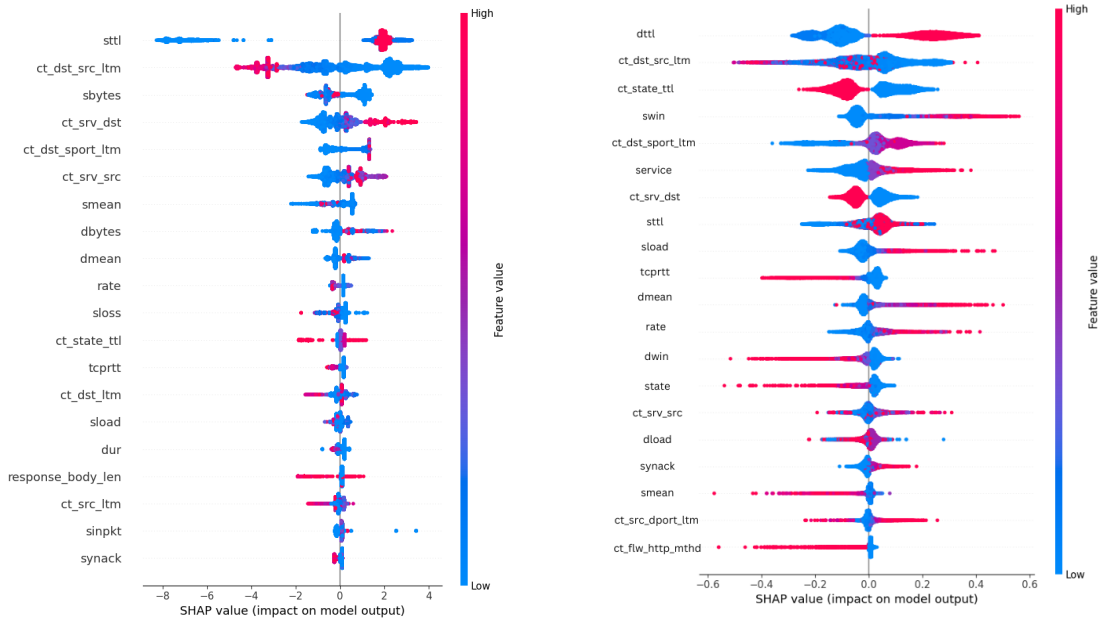
   Similarly, on the **CSE-CIC-IDS2018** dataset, the ANN model demonstrated strong predictive power. It correctly classified **83,883** instances of malicious traffic and **71,557** instances of benign traffic. Despite encountering **8,915** false positives and **2,768** false negatives, the model maintained robust performance in accurately identifying malicious activities, reflecting its adaptability to varying dataset characteristics and complexities.

III.7.3 SHAP Explanation:

Explanation on how a SHAP Beeswarm Plot Works:

The graph produced by the SHAP method can be interpreted as follows [59] :
Features on the Y-Axis: Each row corresponds to a feature in the dataset.
SHAP Values on the X-Axis: The position on the X-axis indicates the impact of the feature on the model's prediction. Values to the right of the center represent positive SHAP values, which tend toward the anomaly class, while values to the left represent negative SHAP values, which tend toward the normal class.
Color Gradient: The color represents the feature value (usually from low to high). Typically, blue indicates low feature values, and red indicates high feature values.
Density of Points: Each dot represents a single prediction (or instance). The density of points (thickness of the swarm) indicates how frequently certain SHAP values occur for that feature.
Interpretation: Features with wider swarms or more spread out points have a

larger impact on the model's predictions. The color distribution within a feature's row shows how different values of that feature contribute to the prediction.



((a)) XGBoost ((b)) ANN

Figure III.5: SHAP Value impact on model output for the (UNSW_NB15)

1. **SHAP Beeswarm Plot Observation for UNSW_NB15**

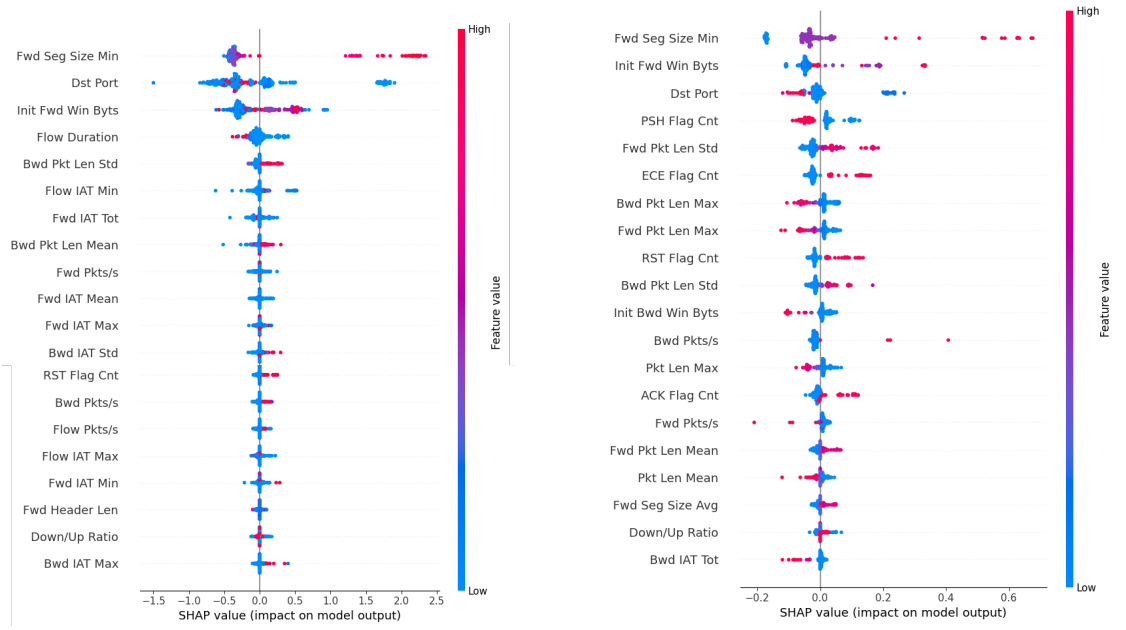
   The Figure III.5 illustrates that feature **ct_dst_src_ltm** (count of destination and source in the last time window) represents the number of times a specific source and destination pair has communicated within a given time frame, it exhibits a wide range of SHAP values, highlighting its strong importance for both **X**GBoost and ANN model. low **ct_dst_src_ltm** values might reflect unusual patterns in destination-source connections over a long-term period, often signaling potential security threats such as scanning attack due to repeated or persistent connections from the same source to various destinations.

   The **sttl** feature shows a significant impact on the XGBoost model predictions, the **sttl** represents the source time-to-live (TTL) value of packets sent by the

source during a communication session. This metric is useful for identifying patterns in network traffic and detecting malicious activities. For instance, TTL-based attacks, where attackers manipulate the TTL values to evade detection or map network structures, can be detected by monitoring unusual patterns in TTL values. An unusually consistent or deviant TTL value from the source can indicate suspicious activities, such as packet spoofing or reconnaissance efforts by an attacker trying to map out the network topology.

The feature **dttl** represents the difference in the TTL values between packets sent and received during a communication session. the SHAP plot of III.5 shows that a high value for **dttl** indicates an attack ,such as TTL spoofing, where attackers alter TTL values to mislead network defenses or trace routing paths.

The feature swin represents the TCP window size offered by the source during a communication session, in the SHAP plot **swin** shows a significant impact on the ANN model prediction, a high value of the feature increase the probability of attack classification, like backdoor attack by using TCP window sizes to communicate with a backdoor in a stealthy manner, avoiding detection by standard network monitoring tools.



((a)) XGBoost ((b)) ANN

Figure III.6: SHAP Value impact on model output for the (CSE-CIC-IDS2018)

2. SHAP Beeswarm Plot Observation for CSE-CIC-IDS2018

Figure III.6 illustrates the most important features that impact the predictions of the ANN and XGBoost models, along with their respective values, as indicated in the following.

Fwd Seg Size Min: (Minimum Forward Segment Size) represents the smallest segment size observed in forwarded packets. High values on the right side of the SHAP plot for XGBoost and ANN indicate a significant impact on the model's output, often signaling malicious activities. This utilization of small segment sizes in forwarded packets, potentially employed to evade detection by transmitting minimal data per packet, this tactic frequently associated with DoS or DDoS cyber attacks.

Init Fwd Win Byts: (Initial Forward Window Bytes) refers to the initial size of the forward window in bytes, which indicates the size of the send window in

TCP connections. High values observed on the right side of the XGBoost plot and ANN plot suggest a potential association with DoS attacks, characterized by using large window bytes to interrupt communications.

Fwd Pkt Len Std: (Forward Packet Length Standard Deviation) This feature represents the standard deviation of the lengths of packets in the forward direction within a flow it observed has high values in the right of ANN and XGBoost plot in this situation represent an attack. In network traffic analysis, it provides insights into the variability of packet sizes being sent from the source to the destination. The variability in packet lengths can be due to different responses from the target system as the attacker tries different credentials this can be happen by bruteForce attack.

Dst Port: (Destination Port) Significant for specific port numbers, it has low values in the left side of the ANN and XGBoost plot that indicate for normal traffic.

III.7.4 LIME Explanation:

To comprehend the model's classification for the given instance, begin by examining the prediction probabilities. Each feature detailed in the table possesses a value pertinent to the data point under analysis. The adjacent bar chart illustrates the extent to which each feature influences the prediction: blue bars signify contributions towards a "Normal" classification, whereas orange bars denote contributions towards an "Attack" classification.

1. **For UNSW_NB15:**

Figure III.8 and III.7 below shows us the Tabular LIME plot for two different models: XGBoost (a) and ANN (b). Both models are applied to the same instance from the UNSW-NB15 dataset to predict whether it is a normal or an attack instance. Let's delve into the specifics of each plot and observe the key differences and similarities.

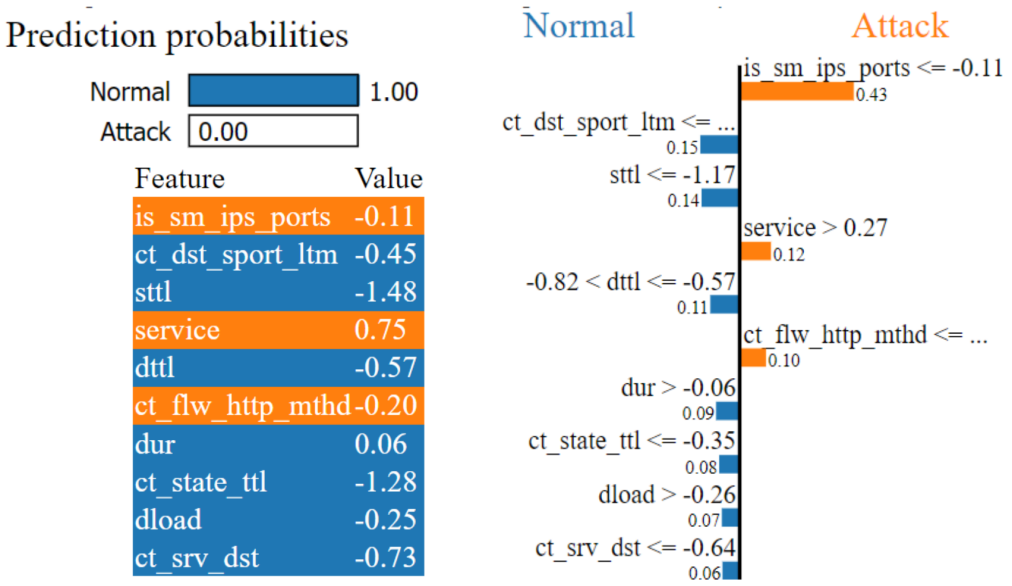


Figure III.7: LIME Explanation Prediction Probabilities for ANN (UNSW-NB15)

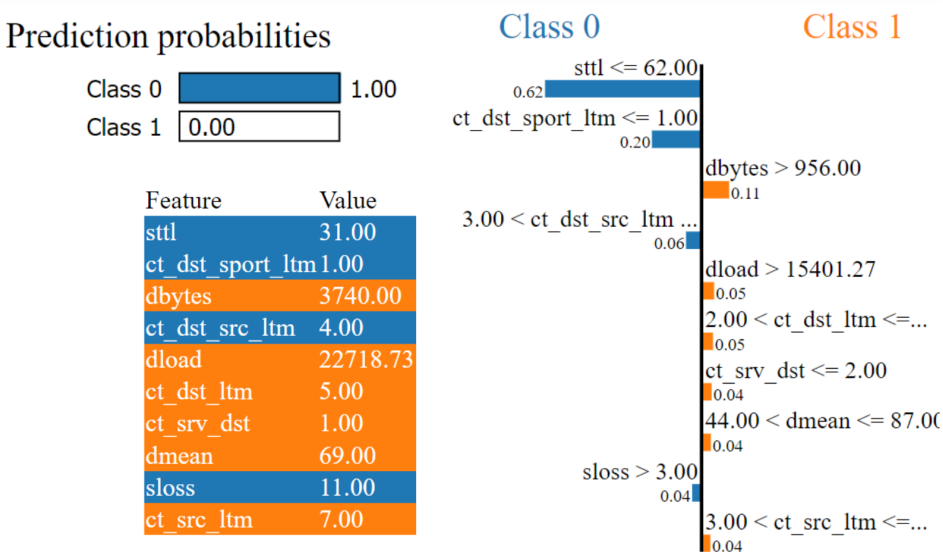


Figure III.8: LIME Explanation Prediction Probabilities for XGBoost (UNSW-NB15)

XGBoost Model

The XGBoost model relies heavily on `sttl` $\leq$ 62.00 and `ct_dst_sport_ltm` $\leq$ 1.00 for its decision, with these features having the highest weights towards classifying the instance as normal.

ANN Model

The ANN model identifies **is_sm_ips_ports** (indicates if the source and destination IP addresses and ports belong to the same subnet) $\leq$ -0.11 as the most significant feature pushing towards the attack classification, however the model still classify as normal, that goes for the other features like **Sttl**, **ct_dst_sport_ltm** represents the count of connections from a specific source port to a particular destination over the last time window and **ct_state_ttl** which represents the count of connections in a specific state with a particular

time-to-live (TTL), there impact collected impact kept the feature get classified as normal.

2. **For CSE-CIC-IDS2018:**

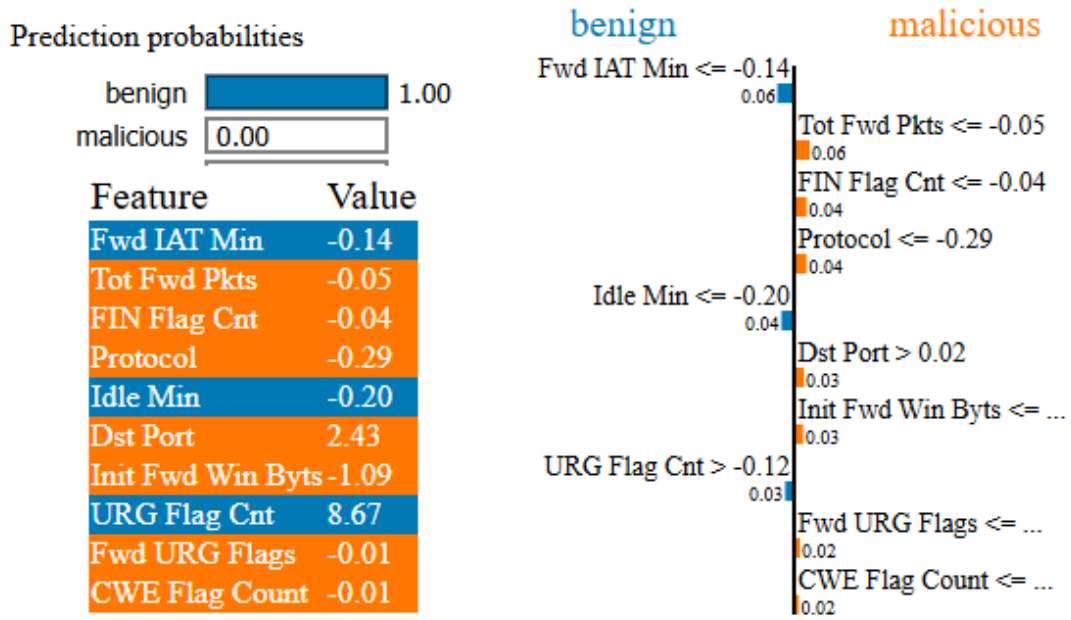


Figure III.9: LIME Explanation Prediction Probabilities for XGBoost (CSE-CIC-IDS2018)

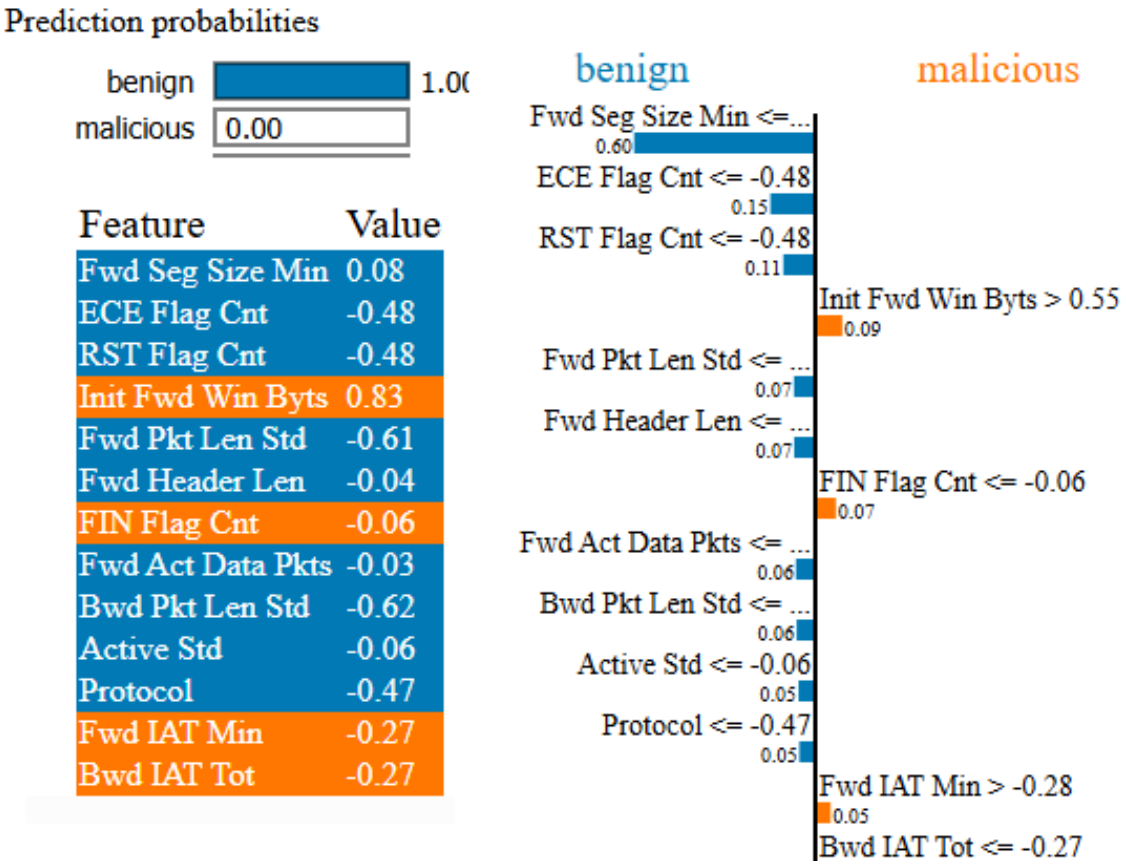


Figure III.10: LIME Explanation Prediction Probabilities for ANN (CSE-CIC-IDS2018)

The XGBoost model for CSE-CIC-IDS2018 shows features like **Fwd IAT Min** (Minimum Forward Inter-Arrival Time) and **Idle Min** (Minimum Idle Time) contributing to benign predictions, while **URG Flag Cnt** (Urgent Flag Count) indicates malicious activity, with high confidence in benign classification (1.00). Similarly, the ANN model highlights features such as **ECE Flag Cnt** (Explicit Congestion Notification-Echo Flag Count) and **RST Flag Cnt** (Reset Flag Count) leaning towards benign predictions, while **Init Fwd Win Byts** (Initial Forward Window Bytes) suggests malicious intent, also with high confidence in benign classification (1.00).

III.7.5 Discussion

Based on experimentation, it can be concluded that the evaluation metrics for both ANN and XGBoost models demonstrate remarkably similar performance across two distinct datasets. Metrics such as accuracy, precision, recall, and F1 score consistently reflect the models' effectiveness in classifying NIDS and HIDS data into benign or malicious categories. The SHAP plots for both ANN and XGBoost models reveal that the impacted features are mostly similar, indicating that both models respond similarly to these features. This similarity is also observed in the LIME explanations, where the features influencing individual instances are largely the same for both models. Comparing the results provided by the Local (LIME) and the Global (SHAP) explanations, it is evident that the features with the highest impact on both ANN and XGBoost models are generally alike. another thing worth to be mentioned is that some types of cyberattacks cannot be effectively observed in the plots due to the scant number of samples in the datasets.

III.8 Conclusion

In this chapter, we implemented the methodology by applying ANN and XGBoost models to the UNSW_NB15 and CIC-IDS-2018 datasets. We conducted data preprocessing and evaluated performance using accuracy, precision, recall, and F1 score metrics. SHAP and LIME were used to interpret model predictions, enhancing transparency. The results highlighted the effectiveness of ANN and XGBoost in different scenarios and the importance of model interpretability in cybersecurity. Our findings suggest that SHAP and LIME are valuable for improving intrusion detection systems, indicating areas for further research and development.

GENERAL CONCLUSIONS

In conclusion, modern technology, particularly AI, ML , DL, XAI, and IDS, has significantly reshaped various domains, especially cybersecurity. This thesis addresses the challenge of the "black box" nature of AI and the need to build human trust in the results generated by these AI models, with a particular focus on IDS. XAI methods were applied to understand the decisions made by IDS based on ML. Our contribution includes the use of two different types of datasets, which are relatively new and encompass both NIDS and HIDS traffic. Two different AI methodologies were utilized: supervised ML XGBoost and supervised DL ANN. Finally, XAI methods, such as local explanations LIME and global explanations SHAP, were applied to make AI models more transparent and to understand why IDS make certain decisions, thereby increasing trust in these decisions.

While this thesis has laid a solid groundwork, several intriguing avenues for further research in the domain of XAI remain:

1. Focus on testing models with more diverse and extensive datasets to validate the generalizability and robustness of results;

2. Developing real-time XAI systems would offer immediate insights into IDS decisions, enhancing their practical applicability in dynamic environments;

3. Using multiclassification AI models and SHAP for explanation enables us to identify different types of cyberattacks with enhanced clarity and precision. This approach allows us to discern specific features and patterns that characterize various cyber threats.

Bibliography

[1] The SDH-V, official website, https://www.the-sdh-v.xgboostProductDetail.aspx?iid=1086970173&pr=39.88/, accessed mai 9, 2024.

[2] W. R. Swartout, Explaining and justifying expert consulting programs, in: Computer-assisted medical decision making, Springer, 1985, pp. 254–271.

[3] B. Tomasik, Artificial intelligence and its implications for future suffering, Foundational Research Institute: Basel, Switzerland (2017).

[4] D. Falk, Q. Magazine, How artificial intelligence is changing science, Quanta Magazine 11 (2019).

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

[6] T. O. Ayodele, Types of machine learning algorithms, New advances in machine learning 3 (19-48) (2010) 5–1.

[7] P. Domingos, A few useful things to know about machine learning, Communications of the ACM 55 (10) (2012) 78–87.

[8] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: European conference on computer vision, Springer, 2016, pp. 69–84.

[9] M. F. A. Hady, F. Schwenker, Semi-supervised learning, Handbook on Neural Information Processing (2013) 215–239.

[10] Y. Li, Deep reinforcement learning: An overview, arXiv preprint arXiv:1701.07274 (2017).

[11] Z. Wu, J. Li, M. Cai, Y. Lin, W. Zhang, On membership of black-box or white-box of artificial neural network models, in: 2016 IEEE 11th conference on industrial electronics and applications (ICIEA), IEEE, 2016, pp. 1400–1404.

[12] W. J. Von Eschenbach, Transparency and the black box problem: Why we do not trust ai, Philosophy & Technology 34 (4) (2021) 1607–1622.

[13] M. T. Ribeiro, S. Singh, C. Guestrin, " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.

[14] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018.

[15] S. Tan, R. Caruana, G. Hooker, Y. Lou, Distill-and-compare: Auditing black-box models using transparent model distillation, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 303–310.

[16] P. Mavrepis, G. Makridis, G. Fatouros, V. Koukos, M. M. Separdani, D. Kyriazis, Xai for all: Can large language models simplify explainable ai?, arXiv preprint arXiv:2401.13110 (2024).

[17] K. Roshan, A. Zafar, Using kernel shap xai method to optimize the network anomaly detection model, in: 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), 2022, pp. 74–80. doi:10.23919/INDIACom54597.2022.9763241.

[18] S. Bhattacharya, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, U. Tariq, A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu, Electronics 9 (2) (2020) 219.

[19] Y. Li, R. Ma, R. Jiao, A hybrid malicious code detection method based on deep learning, International Journal of Security and Its Applications 9 (5) (2015) 205–216.

[20] J. R. Vacca, Computer and information security handbook, Newnes, 2012.

[21] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, T. R. Gadekallu, Federated learning for intrusion detection system: Concepts, challenges and future directions, Computer Communications 195 (2022) 346–361.

[22] S.-W. Lee, M. Mohammadi, S. Rashidi, A. M. Rahmani, M. Masdari, M. Hosseinzadeh, et al., Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review, Journal of Network and Computer Applications 187 (2021) 103111.

[23] Z. Wu, J. Wang, L. Hu, Z. Zhang, H. Wu, A network intrusion detection method based on semantic re-encoding and deep learning, Journal of Network and Computer Applications 164 (2020) 102688.

[24] I. F. Kilincer, F. Ertam, A. Sengur, Machine learning methods for cyber security intrusion detection: Datasets and comparative study, Computer Networks 188 (2021) 107840.

[25] A. Aldweesh, A. Derhab, A. Z. Emam, Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, Knowledge-Based Systems 189 (2020) 105124.

[26] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, Journal of Network and Computer Applications 36 (1) (2013) 16–24.

[27] F. Sabahi, A. Movaghar, Intrusion detection: A survey, Systems and Networks Communication, International Conference on 0 (2008) 23–26. doi:10.1109/ICSNC.2008.44.

[28] M. Wang, K. Zheng, Y. Yang, X. Wang, An explainable machine learning framework for intrusion detection systems, IEEE Access 8 (2020) 73127–73141. doi:10.1109/ACCESS.2020.2988359.

[29] O. Arreche, T. R. Guntur, J. W. Roberts, M. Abdallah, E-xai: Evaluating black-box explainable ai frameworks for network intrusion detection, IEEE Access (2024).

[30] Y. Wang, P. Wang, Z. Wang, M. Cao, An explainable intrusion detection system, in: 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), IEEE, 2021, pp. 1657–1662.

[31] S. Hariharan, R. Rejimol Robinson, R. R. Prasad, C. Thomas, N. Balakrishnan, Xai for intrusion detection system: comparing explanations based on global and local scope, Journal of Computer Virology and Hacking Techniques 19 (2) (2023) 217–239.

[32] H. C. Tanuwidjaja, T. Takahashi, T.-N. Lin, B. Lee, T. Ban, Hybrid explainable intrusion detection system: Global vs. local approach, in: Proceedings of the 2023 Workshop on Recent Advances in Resilient and Trustworthy ML Systems in Autonomous Networks, 2023, pp. 37–42.

[33] M. M. Alani, A. Mashatan, A. Miri, Explainable ensemble-based detection of cyber attacks on internet of medical things, in: 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), IEEE, 2023, pp. 0609–0614.

[34] D. Gaspar, P. Silva, C. Silva, Explainable ai for intrusion detection systems: Lime and shap applicability on multi-layer perceptron, IEEE Access (2024).

[35] XGBoost Documentation, Xgboost documentation, `https://xgboost.readthedocs.io/`, accessed jun 19, 2024.

[36] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.

[37] S. R. Dubey, S. K. Singh, B. B. Chaudhuri, Activation functions in deep learning: A comprehensive survey and benchmark, Neurocomputing 503 (2022) 92–108.

[38] J. Lederer, Activation functions in artificial neural networks: A systematic overview, arXiv preprint arXiv:2101.09957 (2021).

[39] D. Garreau, U. Luxburg, Explaining the explainer: A first theoretical analysis of lime, in: International conference on artificial intelligence and statistics, PMLR, 2020, pp. 1287–1296.

[40] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30 (2017).

[41] C. Molnar, G. König, J. Herbinger, T. Freiesleben, S. Dandl, C. A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup, B. Bischl, General pitfalls of model-agnostic interpretationvrgj4y4i57igtvvvxccxdfggyh7„ dbbv

bfb45ytu6oi53ujuhgyjshrgs.krenvijtkreshqwertyuiop[';/.,mnbvqwertyuiop;.,mnb vcmethods for machine learning models, in: International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers, Springer, 2020, pp. 39–68.

[42] C. official website, https://research.google.com/colaboratory/faq.html: :text=colab

[43] P. official website, https://www.python.org/doc/essays/blurb/Accessed April 2, 2024.

[44] official website, https://www.tensorflow.org/Accessed April 2, 2024.

[45] official website, https://keras.io/Accessed April 2, 2024.

[46] official website, https://docs.anaconda.com/free/navigator/index.html: :text=anaconda

[47] https://domino.ai/data-science-dictionary/jupyter-notebookAccessed April 2, 2024.

[48] L. Liu, G. Engelen, T. Lynar, D. Essam, W. Joosen, Error prevalence in nids datasets: A case study on cic-ids-2017 and cse-cic-ids-2018, in: 2022 IEEE Conference on Communications and Network Security (CNS), 2022, pp. 254–262. doi:10.1109/CNS56114.2022.9947235.

[49] N. Moustafa, J. Slay, Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in: 2015 military communications and information systems conference (MilCIS), IEEE, 2015, pp. 1–6.

[50] L. Truong, C. Jones, B. Hutchinson, A. August, B. Praggastis, R. Jasper, N. Nichols, A. Tuor, Systematic evaluation of backdoor data poisoning attacks on image classifiers, 2020, pp. 3422–3431. doi:10.1109/CVPRW50498.2020.00402.

[51] Fortinet, Types of cyber attacks.

[52] Cloudflare, What is a botnet?, accessed: 2024-06-25.
URL https://www.cloudflare.com/fr-fr/learning/ddos/what-is-a-ddos-botnet/

[53] Fortinet, Brute force attack, https://www.fortinet.com/resources/cyberglossary/brute-force-attack#:~:text=A%20brute%20force%20attack%20is,and%20organizations%27%20systems%20and%20networks., accessed: 2024-06-25.

[54] G. Developers, Classification: Accuracy — Machine Learning Crash Course — Google Developers, https://developers.google.com/machine-learning/crashcourse/classification/accuracy, last accessed 2023-07-23.
URL https://developers.google.com/machine-learning/crashcourse/classification/accuracy

[55] G. Developers, Classification: Accuracy — Machine Learning Crash Course — Google Developers, https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall, last accessed 2023-07-23.
URL https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall

[56] Wikipedia, F-score - Wikipedia, https://en.wikipedia.org/wiki/F-score, last accessed 2021-12-29.
URL https://en.wikipedia.org/wiki/F-score

[57] Confusion matrix — Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Confusion{_}matrix, "https://en.wikipedia.org/wiki/Confusion_matrix, last accessed 24-07-2023".
URL https://en.wikipedia.org/wiki/Confusion_matrix

[58] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (7553) (2015) 436–444.

[59] E. Gürbüz, Ö. Turgut, İ. Kök, Explainable ai-based malicious traffic detection and monitoring system in next-gen iot healthcare, in: 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), IEEE, 2023, pp. 1–6.