School of Engineering and Information Technology

The University of New South Wales - Canberra

Explainable Intrusion Detection Models in Internet of Things

by

Nam Pham

Thesis submitted as a requirement for the degree of

Bachelor of Computing and Cyber Security (Honours)

Submitted:	October 2021	Student ID:	z5196268
Supervisor:	Dr Nour Moustafa	Topic ID:	

Abstract

With the prevalence of the Internet of Things (IoT), human life is becoming more dependent on digital appliances; thus, information technology systems and networks' security and privacy are crucial to protect them against cyberattacks. IoT networks, including various sensors and actuators linked to the Internet, contain a wide range of cyberattacks. Moreover, those Internet-connected heterogeneous devices can generate high-dimensional and multimodal data, hindering cyber defence systems, especially Intrusion Detection Systems (IDS), from detecting and explaining cyberattacks.

Multiple security mechanisms have been proposed to protect IoT networks, and the most prominent approach is anomaly-based IDS. Artificial Intelligence (AI) technology, especially Machine Learning (ML) and Deep Learning (DL), has been utilised for establishing effective anomaly detection methods, which enable them to learn characteristics of malicious behaviours from large-scale datasets. However, most of this type of AI-based IDS uses a black-box model as its decision engine, whose decisions are opaque to users and mitigate their further practical implementation in critical sectors for several reasons. Moreover, the ability to understand and interpret model's decisions can benefit various types of audiences and indirectly improve the performance of cyber defence systems.

This thesis identifies research opportunities to develop an explainable deep learning-enabled intrusion detection framework in IoT networks. This thesis aims to develop an explainable AI-based IDS as an effective cyber defence solution to evaluate its impact on the performance in IoT networks. The proposed framework and model are trained and evaluated using three current IoT network datasets: NSL-KDD, UNSW-NB15 and ToN_IoT.The decision engine of IDS is built upon a deep neural network with optimised parameters, showing high performance on benchmark datasets. Next, the proposed framework is then utilised to interpret the model's decisions, and the generated explanations are evaluated through comparing with the fundamental characteristics of attack classes. Moreover, an architecture for adaptable IDS is built based on this framework's results.

Acknowledgement

To **Dr Nour Moustafa**

For his guidance and encouragement. I have learned a lot from his knowledge and experience. Thank you for being an excellent supervisor and help me on this journey.

To **Dr Benjamin Turnbull**

For his caring and support. I appreciate the extra effort that he has put on helping me throughout this tough year.

To **Professor Albert Zomaya**

For his mentorship and guidance. I have learned so much from him, and I appreciate the extra effort that he has put in reviewing my paper.

To **Quang Tran**

For his moral support and advice. I am grateful for the encouragement and experience that he gave me when I needed it.

To **Khue Nguyen, ADFA juniors, family and friends**

For the encouragement and love. I would not be where I am today without your invaluable support.

Publications and Presentations

List of Publications

- **Pham, N**; Moustafa, N; Zomaya, A. Explainable Cyber Defences in the Internet of Things Networks: Opportunities and Solutions, Journal of IEEE Communications Surveys & Tutorials, (under review), 2021
- **Pham, N**; Moustafa, N; Turnbull, B. An Explainable Deep Learning-enabled Intrusion Detection Framework in IoT networks, Future Generation Computer Systems, (under review), 2021

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
DL	Deep Learning
IDS	Intrusion Detection System
IoT	Internet of Things
ML	Machine Learning
MSA	Multi-Stage Attacks
XAI	eXplainable Artificial Intelligence

Chapter 1

Introduction

1.1 Introduction

With the rapid growth of information systems and computer networks, human life quality is improved, but we also have several security concerns involved with a wide range of vulnerabilities and cyberattacks against those systems [1]. Securing the Internet of Things (IoT) systems are more challenging than protecting traditional systems due to its heterogeneous nature and ubiquity deployment as a distributed network [2]. IoT incorporates many physical devices, like sensors and actuators, into the Internet to provide automated services to end-users and organisations [3]. Moreover, various connected devices within IoT networks can generate high volume, variety and high speed data that are difficult to analyse and detect malicious activities. This nature and the rapid development of new attack techniques hinder the use of traditional defence mechanisms such as anti-malware firewalls, authentication and encryption. Therefore, Intrusion Detection System (IDS), especially Artificial Intelligence (AI)-based IDS, is a prominent approach that can identify diverse cyber attacks and even zero-day attacks in IoT networks. Due to the application of AI techniques, AI-based IDSs can achieve good performance with benchmark datasets. In AI, deep learning techniques can analyse complex data and learn from previous attack patterns in the dataset to detect zero-day attacks. However, such techniques suffer from high false rates and opaqueness to end-users to understand the predictive outputs of the models [4].

Intrusion Detection System (IDS) is a prominent cyber defence control used extensively to detect cyberattacks effectively. IDS was first introduced by Denning [5] in 1987. AI-based IDS, also known as anomaly-based IDS, is widely-used due to its ability to perform well with large-scale data and detect unknown attacks due to its AI-enable detection engine. Moreover, the need to explain the functioning and predictions of AI-based IDS arises as different types of users are benefited from understanding the root cause of intrusion detection. Therefore, the field of explainable artificial intelligence (XAI) has been developed in recent years to address this concern. XAI illuminates black-box models of ML by providing explanations on their functioning and predictions [6, 7].

This thesis provides fundamental research and scientific knowledge for developing an Explainable AI-based Intrusion Detection System (IDS) that is effective and adaptable to novel and ever-changing attack vectors in IoT platforms. This is achieved by designing an explainable IDS framework to assist developers and researchers in evaluating the quality, analysing and detecting any bias in the training set of AI-based IDS. The proposed framework promotes the interpretability of black-box models for the detection engine of anomaly-based IDS, providing a basis for developing robust explainable AI-based IDS in cyber defence systems. Moreover, an effective, flexible and explainable IDS is proposed that utilise this framework's explanation of the model's predictions. Throughout this research, we use a data-driven problem-oriented approach to identify the research gap in this field, then propose solutions to interpret the model's predictions and improve the performance of IDS in IoT networks.

This thesis explores the field of Intrusion Detection System (IDS) for Cyber Defence, with a focus on eXplainable AI (XAI) and cyber threats in Internet of Things (IoT). The research opportunities are then identified through the analysis of existing works in these fields, and solution are designed based on a data-driven approach to enhance the performance of existing literature [4, 6, 7] and design a framework for explainable

AI-based IDS. An explainable AI-based IDS in IoT networks is established, and an architecture of an effective, flexible and explainable IDS is proposed and implemented. Methodological limitations of the proposed solutions and future research directions are also discussed.

The remainder of this chapter is structured as follows. Section 1.2 discussed the motivations behind this research. Section 1.3 summarizes the research challenges and questions that will be addressed throughout this thesis. Section 1.4 presents the structures for the remainder of this work.

1.2 Motivation

The research motivation for this thesis comes from several aspects. The first is the identified research gap in the development of explainable AI-based IDs in heterogeneous IoT platforms [7]. The application of AI, especially Machine Learning (ML) and Deep Learning (DL) techniques in IDS, is promising; but black-box models remain opaqueness to users and operators, hence hindering its application in real-life deployment in critical sectors [8]. The aim of XAI is to illuminate such black-box models and enable AI-based IDS for further applications. Secondly, we need to identify the most suitable set of XAI methods to be used specifically in IDS, whose explanations are intuitive and easy to understand by various types of audiences [9]. The finding of the most effective XAI methods will enhance the benefits of XAI as recipients can comprehend and utilize the generated explanations. Thirdly, the explanations produced by the XAI framework presents an exciting research direction: to use the generated outputs effectively to enhance the interpretability and flexibility of existing IDS. XAI can promote the applications of AI techniques by improving the capability of model debugging, data collection, human decision-making and trust-building. Therefore, developing an effective, flexible and explainable IDS based on the XAI framework is promising.

1.3 Research Question

This thesis provides solutions to the following research questions:

Research Questions: *How could we design an explainable AI-enabled IDS that is effective and adaptable to ever-changing in IoT environments?*

This question is decomposed into the following sub-questions:

Sub-question 1: *How robust are existing explainable AI methods in interpreting decisions of deep learning-enabled IDS?*

Sub-question 2: *How could we build an explainable AI-enabled IDS in IoT networks?*

Sub-question 3: *Which architectures ensure IIDS are effective, explainable and adaptable in IoT networks?*

1.4 Thesis Structure

This thesis is structured as follows. Chapter 2 explores the existing literature in the research field relevant to this work. To be more specific, Chapter 2 discusses the current state of Intrusion Detection Systems, Artificial Intelligence for Cyber Defence, Internet of Things platform, and AI-based Cyber defences, focusing on Explainable AI-based IDS. This chapter collects recent works in these fields and compares them to find each approach's advantages and disadvantages. Therefore, Sub Question 1 is addressed in this chapter, which analyses existing XAI methods and their ability to interpret the decision of black-box models. Also, in this chapter, the research opportunities are identified, and their necessity is proved. Next, Chapter 2 concludes with the research challenges and future research directions of the above field.

Chapter 3 and 4 addresses sub-question 2 and 3, respectively. Each chapter addresses a phase of the research, and each contains independent methodology, design, results, analysis and discussion. Chapter 3 focuses on designing an explainable Deep Learning-enable Intrusion Detection framework in IoT networks in which explanations are intuitive and easy to be used by various types of XAI recipients. Next, Chapter 4 proposes a design of Intrusion Detection System build upon the framework in Chapter 3. The novel design is proved to be effective, explainable and adaptable to heterogeneous IoT environments. Finally, Chapter 5 concludes this thesis with a summary of research, contribution to knowledge, methodological limitations and future research direction of this work.

Chapter 2

Literature Review

2.1 Introduction

As human life is becoming more dependent on digital appliances, the security and privacy of information technology systems and networks are crucial to any organisation [1]. The field of cybersecurity was developed to address this concern. Internet of Things (IoT) networks, including sensors and actuators, would improve human life quality, but they contain a wide range of vulnerabilities for malicious purposes [1]. Moreover, due to the proliferation of heterogeneous devices, IoT networks generate high-dimensional and multimodal data, which requires analysing big data. Artificial Intelligence (AI) can attain this requirement since AI technologies, especially Machine Learning (ML) and Deep Learning (DL), have been utilised across industries and achieve excellent performance with large scales of data.

Due to the potential benefits of compromising computer systems, malicious actors invest a lot of money, time, and effort in making sophisticated cyber attacks. Zero-day and multi-stage attacks are significant challenges to ensuring digital assets' security as they

[1] The work in this chapter has produced this paper (**Pham, N**; Moustafa, N; Zomaya, A. Explainable Cyber Defences in the Internet of Things Networks: Opportunities and Solutions, Journal of IEEE Communications Surveys & Tutorials, (under review), 2021)

can bypass traditional security mechanisms. While zero-day attacks utilise unknown techniques or exploits, multi-stage attacks combine multiple stages in which each step taken is insufficient to be recognised as malicious [10]. Advanced Persistent Threats (APTs) would use even more complex techniques to achieve their malicious intent [11]. Moreover, due to the heterogeneous and resource-constraint nature, IoT networks can introduce many vulnerabilities and make it challenging to discover attacks as well as implement security mechanisms.

To prevent malicious activities and protect the system, multiple security mechanisms have been proposed. Traditional approaches, such as anti-malware, firewalls, user authentication and data encryption, are all well-known, and each of them fits different purposes [12]. However, traditional mechanisms are ineffective due to lacking dynamism and the rapid growth of attack techniques [13]. An IDS is a prominent technology to identifying diverse cyber attacks and even zero-day attacks. As a result, recent years have witnessed many advancements in IDS. Due to the application of AI techniques, AI-based IDS can achieve good performance with benchmark datasets. Deep learning algorithms can analyse complex data and learn from previous attack patterns in the dataset to detect zero-day attacks. However, such techniques suffer from a high false-positive rate and opaqueness to users.

Anomaly-based detection is important to prevent unknown attacks and protect information technology systems. Therefore, many ML algorithms have been proposed to design efficient IDS with high accuracy and low false-positive rates. Moreover, the need to explain the functioning and predictions of ML-based IDS arises as different types of users are benefited from understanding the root cause of intrusion detection. Therefore, the field of explainable artificial intelligence (XAI) has been developed in recent years to address this concern. XAI illuminates the black-box model by providing explanations on their functioning and predictions.

Motivation – The challenges to achieving XAI for anomaly-based IDS are a combination of difficulties in developing an effective anomaly-based IDS and obtaining explainability in AI. On the one hand, the challenges for designing effective IDS mainly consist of a comprehensive dataset and real-time detection. The dataset is crucial in anomaly-based IDS as it is used to train the ML model and significantly affect the model's performance in real-life deployment; however, collecting a comprehensive and good-quality dataset that reflects all possible types of intrusion is impossible [1, 9]. Moreover, real-time detection capability is challenging to achieve as it cause a long processing time and high false alarm rate [7, 14]. On the other hand, designing an efficient XAI method that can be easily accessible and evaluated is difficult. XAI needs to generate explanations for multiple types of the recipient so that the model's predictions can be thoroughly understood; however, existing literature only focus on a few types of audience such as researchers and developers [15]. Moreover, the lack of standardised terminologies and definitions in the field causes difficulties in the evaluation model's performance [16].

Contribution–This chapter presents a comprehensive review that discusses XAI methods and techniques for developing explainable cyber defences, especially anomaly-based IDS, in IoT networks. The major contributions of this chapter include the follows:

1. We demonstrate intrusion detection systems with a focus on anomaly-based detection in IoT networks. Different types of IDS are categorised and analysed. Their utilisation of these systems for various IoT environments are discussed, demonstrating their advantages and disadvantages.

2. We review AI techniques, including machine learning and deep learning, and explain their types. We also explain how these techniques can effectively learn from large-scale datasets to discover and explain cyberattack events

3. The field of XAI and its utilisation for anomaly-based IDS is discussed in-depth. In regards to the need for XAI and its benefits across various sectors, we explore various existing XAI approaches and their potential applications for anomaly-based IDSs in IoT networks.

4. We survey recent studies in the intersection of XAI, anomaly-based IDS and IoT. We show insights into the intersection of those topics and analyse each piece of research.

5. We also identify current challenges and provide future research directions related to XAI for cyber defences in IoT networks. The application of XAI in cybersecurity presents several challenges and provides new opportunities for further research to develop explainable AI-based IDSs in current IoT networks.

This literature review consists of seven sections. Section 2.2 outlines cybersecurity and terminologies in this field. Section 2.3 discusses Intrusion Detection System and Network Anomaly-based Detection System. Section 2.4 discusses Artificial Intelligence and its role in cybersecurity along with Explainable Artificial Intelligence. Section 2.5 introduces the Internet of Things and the security challenges to securing the IoT networks. Section 2.6 discusses and summarises related work. Lastly, Section 4.5 concludes the literature review and summarise the information provided.

2.2 Overview of Cyber Security

Over the last decades, there has been an immense increase in the use of computing and digital appliances. People utilize these appliances as they provide convenient and effective means of communication. Ultimately, these devices connect the virtual worlds with the physical worlds, increasing efficiency. Consequently, humans' daily lives heavily depend on these computing networks, applications, or devices [12].

As people become more dependent on the Internet and digital appliances, security issues become more popular and lead to more severe consequences. According to Cybersecurity Ventures, global cybercrime costs grow by 15 per cent per year and will reach $10.5 trillion USD annually by 2025 [17]. Over the period 2020-2021 financial year, the Australian Cyber Security Centre (ACSC) received 67,500 cybercrime reports at an average of one report every 8 minutes. A total of more than $33 billion (AUD) are self-reported to get lost due to cybercrime [18]. This explains why many countries are expected to spend billions of USD for securing the information systems. According to Atlas VPN investigation, the US government is expected to spend $18.78 billion for cybersecurity in 2021 [19].

As mentioned previously, the privacy and security of information systems are of utmost need to any organization. The term cybersecurity is defined in different ways due to the number of different aspects. In [20], the authors defined *cybersecurity* as the security and privacy of digital assets, which are everything from computer networks to mobile devices and data that is processed, stored, and transferred by interconnected information systems. Cyber security's goal is to preserve the integrity, confidentiality, and availability of information in cyberspace [20].

In order to comprehensively define cybersecurity, the term cybersecurity refers to the processes, guidelines, technologies, and practices of defending cyberspace from malicious activities. Cyberspace is a global domain in which electronic and electromagnetic spectrum combined with interconnected and dependent networks help create, edit, store and transfer information [21, 22]. Cybersecurity strives to detect and prevent security risks in the cyber environment by following general security objectives, including confidentiality, integrity and availability (CIA triad) [23].

Some terminologies in cybersecurity are as follows:

- **Vulnerabilities** are flaws or weaknesses in a system that attackers can exploit by executing malicious commands, accessing data without having authorization, or conducting denial-of-service attacks [24] [25]. More generally, vulnerabilities refer to any components of the information system that is exploitable and threatens the whole system's security.

- **Cyber Threats** are actions that can be taken from existing vulnerabilities in a system to gain benefit [26]. Different from vulnerabilities, threats would involve outside elements.

- **Cyberattack/intrusion** is a set of intentional actions taken to exploit an information system using different techniques to compromise confidentiality, integrity, availability of the system and achieves malicious goals [27] [28]. Methods to launch a cyber attack can be malware, phishing, SQL injection, Man-In-The-Middle (MITM) and many others.

- **Zero-day attacks** can be variants of known attacks [29] or unknown attacks that exploit zero-day vulnerabilities in the system. Zero-day vulnerabilities are unknown to the public and are not yet discovered by software vendors or network defenders.
- **Multi-stage attack (MSA)** is a type of intrusion consisting of a sequence of correlated techniques [10]. The MSA utilizes more complex attack techniques over a long period [30] in which each step taken is insufficient to be recognized as aggression. Steps in a typical MSA can be classified into different phases, including reconnaissance, weaponization, delivery, exploitation, installation, command, and control, actions on objective regarding Lockheed Martin's Cyber kill chain [31].
- According to the National Institute of Standards and Technology (NIST), **Advanced Persistent Threats** (APTs) are adversaries which ultimately aim to steal information, undermine or impede critical aspects of a mission, program, or organization, or place itself in a position to do so in the future. To do that, APTs establish and extend their presence in information systems by using a wide range of attack surfaces with sophisticated levels of expertise and significant resources. APTs typically launch the attack through multiple steps; they are persistent, targeted attacks on a particular organization [32]. Therefore, APT attacks are a complex version of multi-stage attacks.

2.3 Intrusion Detection System

Due to the heavy reliance on information systems, various approaches have been deployed to protect information systems such as access control, firewall, anti-malware, sandbox and cryptography [12]. Those are traditional mechanisms that have been widely adopted nowadays; however, due to the rapid growth of attack techniques and zero-day vulnerabilities, such methods are insufficient for securing information system. Thus, an IDS, which is the main method capable of preventing a wide range of cyber-attacks and even zero-day attacks, has become necessary in any organisation's

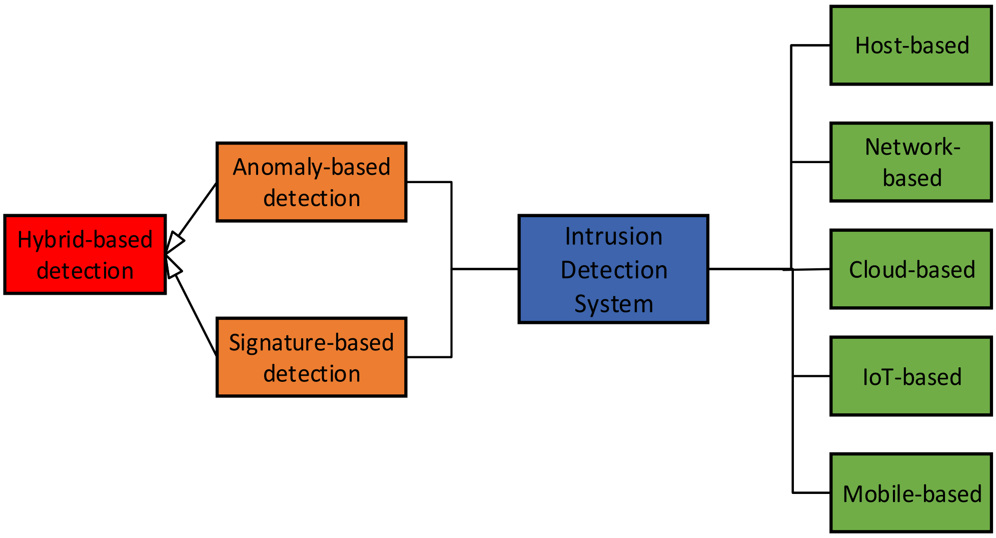


Figure 2.1: Classification of IDS

security infrastructure. *Intrusion detection* is defined as *"the process of monitoring the events occurring in a computer system or network and analysing them for signs of intrusions"* [33]. Bace and Mell (2001) described the intrusions as attempts to compromise the confidentiality, integrity and availability or exploit the security mechanism of digital assets [33]. The aim of IDS is to observe traffic and identify possible threats in the network and computer system by supervising, identifying and evaluating their violations of the security principles [34, 35].

Based on the place of deployment of IDS and the type of system the IDS protects, multiple types of IDS have witnessed a lot of research effort:

- Host-based IDS (HIDS) is attached to the operating system kernel of a specific host and protects the host by forming a layer that allows only legitimate system calls to go through. In other words, HIDS is attached to a single host and watch for malicious activities. In HIDS, computational resources that power the host-based system are taken from the attached host. Host-based methods are reactive, which means that they alert the host after an attack has occurred [36]. Moreover, it might be exposed if the host server is compromised [37] and it cannot be compatible with different platforms [38]. However, HIDS is useful for identifying malicious activities in the organisation's internal pieces of equipment by monitoring system calls, processes, file-system changes and application logs [39]. Moreover, it can analyse the encrypted or obfuscated payloads in the network [37].

- Network-based IDS (NIDS) monitor and analyse activities in the network by reading all inbound packets in the entire network that it is deployed [40]. NIDS do not need to use system resources [36] and can monitor network activities over a particular network segment regardless of the type of the operating system [41], which makes NIDS portable. However, one major drawback of NIDS is that it cannot process encrypted or obfuscated payloads since it only captures information from packet headers [42].

- Cloud-based IDS typically has several different places of deployment. Host-based IDS can be deployed in the virtual machines (VMs) hosted on a cloud server or placed in the hypervisor to monitor the traffic within the hypervisor and the information transferring between the VMs on that hypervisor. Network-based IDS can be deployed to detect abnormalities in the virtual network traffic or to monitor unencrypted network traffic between virtual machines [43].

- IoT-based IDS can be classified into two approaches based on the IDS place of deployment. Centralised IDS is the most widely-used [44] method in which a dedicated central unit such as cloud is utilised to monitor and analyse the traffic data in IoT network. Centralised IoT-based IDS offer the advantages of mighty computation power and centralised management of network traffic; however, it degrades the network performance by generating significant communication overhead [45]. Due to the increasing data traffic that IoT devices generate, this method is gradually becoming unsuitable and replaced. To tackle those problems, decentralised IDS distributes the centralised computing and computational tasks to local fog nodes. Therefore, the heavy load of monitoring traffic data is decreased, and the processing capacity is increased [46].

- Mobile-based IDS has gain popularity in recent years due to the emergence of smartphones. Mobile-based IDS can be further classified into three categories, including (1) host-based: the IDS is deployed on the mobile device, (2) centralised: IDS that is deployed within the cloud will monitor and analyse the mobile devices, (3) distributed: IDS is partly deployed on the cloud and partly on the device [47]. Cloud computing allows centralised data collection and processing; thus, it is convenient and practical to deploy IDS on the cloud to utilise the powerful computation power and memory capacity. However, relying on a cloud server has two limitations, including continuous connectivity to the central server and the risk of sensitive information leakage [48].

An IDS can be designed based on three detection methods:

- Signature-based detection recognises possible intrusions by comparing patterns against captured signature of known attacks. Each type of attack usually contains a specific pattern called signature [49]. The signature-based detection method relies on the database consisting of the signature of the existing attack to detect them. As a result, all the existing attacks stored in the database would be detected with high confidence. However, zero-day attacks or even variants of known malicious behaviours would easily bypass the signature-based IDS. Moreover, the database containing the signature of known attacks requires to be updated repeatedly by network security experts; otherwise, this method is not effective [50].
- Anomaly-based detection monitors the everyday activities of network traffic to construct a baseline of unmalicious behaviours. The features or patterns that the anomaly-based IDS model would be static or dynamic can be everything developed by counting the number of packets sent, number of failed attempts to log in, and many others [49]. Whenever any activities have deviated from the constructed normal baseline, an alarm would be generated to alert the admin [51]. Therefore, if the baseline of normal behaviours is built carefully and comprehensively, anomaly-based IDS can detect any types of attacks, including both known and zero-day attacks.
- Hybrid-based detection combines both anomaly-based detection and signature-based detection. Therefore, this method of detection can have advantages from

both approaches above. The hybrid-based method depends on the signature-based module to detect known attacks, hence lowering the false alarm rate. At the same time, the anomaly-based module constructs the baseline of normal behaviours in network traffic; thus, it helps detect zero-day attacks.

With the increasing complexity in intrusion techniques, signature-based IDS would be bypassed as attackers utilising zero-day attacks. Anomaly-based IDS can detect zero-day attacks without prior knowledge, which is a significant advantage over the signature-based model. Thus, this thesis will focus on network anomaly-based IDS to build an efficient explainable AI-based IDS with high detection accuracy and low false-positive rate.

2.3.1 Network Anomaly Detection System (NADS)

As mentioned earlier, anomaly-based IDS is effective in detecting known and unknown attacks. It constructs a baseline normal behaviour profile for the monitored network then uses this baseline for comparison of actions at any given time, and the anomalies are reported by raising an alert [1]. A NADS consists of four components: a dataset, data processing module, decision engine (DE) method, and defence responses [52]. A realistic dataset of networks is crucial for building an efficient IDS because it helps to improve the detection accuracy of the IDS in real life and evaluate the performance of the model after training [53].

Building an efficient anomaly-based detection method requires a high-quality data source. However, acquiring a high-quality dataset is challenging due to the difficulty of labelling normal and attack behaviours in live network traffic [54]. Network dataset is collected in a real-time or offline data collection [14]. Then, tools and techniques are employed to store and handle a network's big data [55]. For example, to extract network features, several tools are utilised, such as tcpdump, Zeek (previously known as Bro), and MySQL Cluster CGE [14]. The tcpdump can sniff packets on the network, then Zeek extracts the flow-based features from different protocol types in the pcap

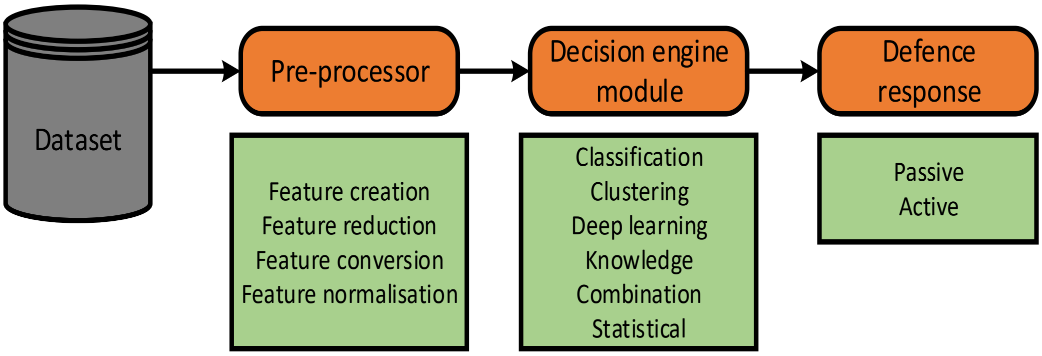


Figure 2.2: Components of NADS

files. After that, MySQL database is used to store all collected features, and then each record is labelled normal or abnormal [14].

After data collection, there are probably many redundant or duplicated records in the dataset. Therefore, the data processing module is essential in improving the performance of an IDS as it removes noisy and irrelevant information from the collected network data in the dataset [14]. Data-processing consists of the creation, reduction, conversion and normalisation of feature [14].

DE module is the most critical component of an IDS. In NADS, DE approaches are classified into six categories, including classification-based, clustering-based, deep learning-based, knowledge-based, combination-based, and statistical-based [14]. In all of the mentioned types of DE modules except knowledge-based and statistical-based approaches, there have been attempts to apply machine learning algorithms to distinguish normal and attack events. The performance of the DE module significantly relies on the quality of the data source; however, constructing a comprehensive normal behaviour baseline profile is impossible in real life. Therefore, the anomaly-based detection model suffers from high false-positive rates due to the lack of normal activities profile in the dataset [56].

Defence responses are actions taken after an attack is detected. There are two types of responses: passive and active. In passive response, the network administrator must take action after the IDS raise an alert about malicious behaviour. The form of alarm would be a popup window or an onscreen alert [14]. The active responses refer to a set of actions taken automatically by the IDS that changes the behaviour of an intrusion, such as disconnect users or terminate connections and attacks [57].

2.4 Artificial Intelligence for Cyber Security

Artificial Intelligence (AI) is a branch in computer science referring to algorithms that simulate human intelligence in machines capable of imitating human behaviour. In the last decade, the field of AI has seen such rapid growth that AI-based algorithms have been developed in every sector of the technology industry and transformed the way we approach real-world tasks. The advancements in machine learning (ML) and deep learning (DL) are leading this growth.

Especially in the field of cybersecurity, ML/DL techniques offer powerful tools in cybersecurity defence that serve multiple purposes. The use of ML/DL methods in malware or intrusion detection and classification has become common. ML can generalise to never-seen-before malware families, and polymorphic strains [58]. Similarly, it allows anomaly-based IDS to detect zero-day attacks. Tuor et al. [59] use DL techniques for insider threat detection by analysing system logs to detect malicious activities. Additionally, multiple ML methods aid cybersecurity forensic by classifying file fragments [60] and detecting kernel rootkits in Virtual Machines (VMs) [61]. Moreover, Sarker et al. [12] propose a security intelligence modelling using the combination of various AI methods and other techniques. The modelling can be used in multiple domains of cybersecurity to protect against phishing attacks and malicious code.

2.4.1 Machine Learning

Machine learning (ML) is a subset of AI which builds a mathematical model based on training data in order to make decisions or predictions without being explicitly

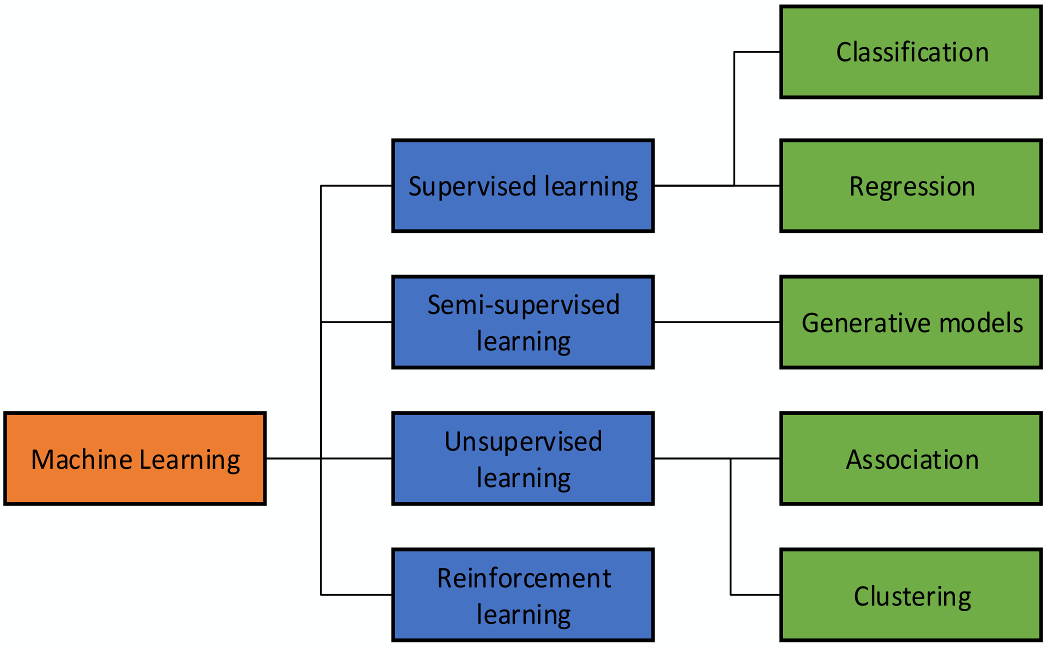


Figure 2.3: Machine Learning Tree

programmed to perform the task [62]. ML models can perform a wide range of tasks in various fields, including cybersecurity. ML's popular real-world applications include image classification, object detection, natural language processing, malware filtering, and many others. Based on the nature of training data or the learning techniques, ML algorithms can be classified into different types as shown in 2.3.

- **Supervised learning**: the training data has labelled input and their desired outputs. After being trained, the ML model can capture the relationships and dependencies between the prediction output and the input features, hence giving correct labels for the features of unknown samples. Supervised learning problems group into classification and regression problems as shown in Figure 2.3.
- **Unsupervised learning**: the training set only includes inputs without the desired output. The ML model extracts and learns the relationship and patterns in data from unlabeled data on its own. Specifically, the system arranges data into categories or clusters from the offered training figures and input patterns. There-

fore, it is also considered as self-organizing, and adaptive learning [63]. There are two main types of unsupervised learning problems, including clustering and association problems.

- **Semi-supervised learning**: this is a combination between supervised learning and unsupervised learning in which the training set includes both unlabeled data and labelled data. Semi-supervised learning is helpful in many scenarios when data collecting is expensive, time-consuming or even unrealistic [64]. This approach can utilize the unlabeled data to improve the learning accuracy [64].

- **Reinforcement learning**: In reinforcement learning, there is no training dataset. The model is trained in a dynamic environment where it interconnects with surrounding by employing trials then receive rewards or penalties depending on its actions. The data feedback is crucial for the model to learn from experience and improve performance. This type of learning is motivated by behaviourist psychology [63].

2.4.2 Deep Learning

Deep learning is a subset of machine learning whose architecture is motivated by the structure and functioning of the human brain. Deep learning architecture is multi-layer neural networks. The network consists of multiple layers constructed and connected through neurons. Each neuron is considered a basic computational component, and the whole network presents the computation of the learning process [65]. Several neurons, which usually equals the number of input features, constitute the input layer of the network [66]. The output layer consists of many neurons that equal the number of different classes in the dataset. However, there is usually only one neuron in the binary classification problem, which is a real number in the range 0-1. The last type of layer is hidden layers that are placed between the input layer and output layer. Deep learning models have multiple hidden layers in their network. In [67], the authors classify deep learning models into two categories based on the architectures as follows:

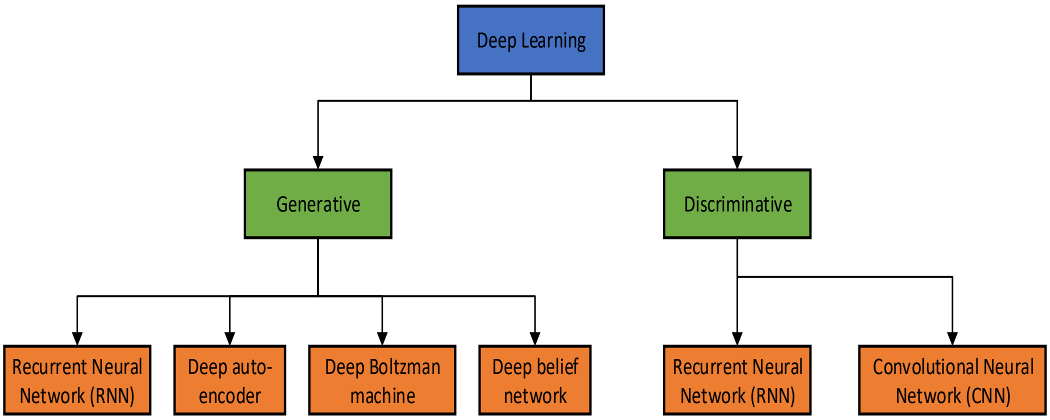

Deep Learning
Generative
Discriminative
Recurrent Neural Network (RNN)
Deep auto-encoder
Deep Boltzman machine
Deep belief network
Recurrent Neural Network (RNN)
Convolutional Neural Network (CNN)

Figure 2.4: Classification of Deep Learning techniques

- **Generative**: The unsupervised learning technique is applied to learn from unlabelled data. Generative models depict independence/dependence for distribution by computing join probability distributions from data with their labels [14]. Models, which can utilize generative architecture, are Recurrent Neural Network (RNN), Deep Auto Encoder (DAE), Deep Boltzmann Machine (DBM) and Deep Belief Network (DBN).

- **Discriminative**: The supervised learning technique is applied to distinguish patterns for prediction tasks. Discriminative models directly estimate the posterior distributions of classes conditioned on the input data [67]. Thus, the discriminative approach is more efficient since it only focuses computational resources on a given task, which is classification, without modelling underlying probability distributions. There are two types of discriminative architecture, including RNN and Convolution Neural Network (CNN).

Generally, in deep learning, data is demonstrated as a nested hierarchy of concepts within the multi-layer neural network. Thus, deep learning can learn the computational process in depth [14] and achieve good performance and flexibility that outperforms the traditional machine learning in data with high scale [68]. In the scope of this thesis, existing IDS datasets that consist of a vast number of records would be used; thus, deep learning approaches are the most suitable to be utilized for building an efficient

IDS.

2.4.3 Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) is an extension of the AI field in which methods are invented to explain ML models' predictions or make models interpretable. The two terms "interpretability" and "explainability" are often used interchangeably by researchers [69]. The field of explainability all started since the publication of paper [70] in the 80s; however, it observed less noticed as AI's growth has focused on predictive performance. In recent years, researchers tend to pay more attention to the XAI topic since AI has involved many critical sectors. In [6], Miller et al. defined "interpretability" as the degree to which a person understands a decision or prediction of any problem in any field. It is worth noticing that a person might be an expert or a normal person with less or even no prior knowledge in the field.

The term black-box model refers to models whose internal designs are secret or cannot be revealed. Regarding the field of AI, the machine learning model is an example of a black-box model that takes input in the form of image, text or tabular data and produces output without any explanations. XAI aims to illuminate these black-box models by making the models interpretable or explaining their predictions. The term responsible AI is defined in [71] as AI models that consider values, moral and ethical concerns. In [71], the authors also propose the ART principles for responsible AI consisting of **Accountability, Responsibility and Transparency**. XAI is the next generation of AI technologies [69] as it shifts AI's development towards designing more robust models, which takes ART principles into account.

2.4.3.1 The need for XAI and its benefits

The ML model is a black-box model; thus, it prevents the development of responsible AI and becomes a barrier to applying AI in further practical implementation. The advancement of such black-box models compromises ART principles and leads to problems such as unethical use, lack of responsibility and accountability, and potential biases in making decisions. Moreover, research in XAI is in utmost need due to laws recently applied by the governments [72]. Arrieta et al. (2020) defined two causes that make the incapability of explaining decisions such a barrier that AI is facing [73]. First, in some critical sectors, relying on black-box models to make predictions is impossible due to the enormous gap between the research community and the business operator. Such sectors have such strict regulations, and the decisions to be made daily are so crucial that operators cannot take risks by trusting a model to give vague predictions. Second, the desire to acquire knowledge and improve understanding promotes the development of XAI. The reason is that every field benefited from AI; not only are the results important but also the ability to understand and explain the results [73].

In [69], XAI's benefits are classified into four different categories, including explain to justify, control, improve and discover. To be more specific, Fig 2.5 presents the benefits of XAI in a way in which the type of audience is the finest aspect. Firstly, it helps people affected by the model's decisions to be aware of their situation. Thus, they truly understand the decision to comply or even disagree with it. Secondly, XAI benefits experts and users as it provides knowledge and gains trust from them. Thirdly, managers would find it easier to assess regulations with XAI's explanations. Fourthly, regulatory agencies can assess the predictions and explanations to decide whether the model complies with the regulation in force, audits. Lastly, XAI is most useful for developers by helping them ensure and improve the model's efficiency. Data scientists are also benefited from XAI as it provides information for them to create or collect new features to improve performance.

In other words, the appearance of XAI methods can enhance the use of ML techniques for application in different industries by promoting the following benefits:

- **Model Debugging**: During the training process, ML models would take biases from the training dataset, which cannot be preventable. Thus, they tend to provide discriminated decisions against underrepresented groups [74]. XAI can present the patterns constructed by the model, which helps data scientists and

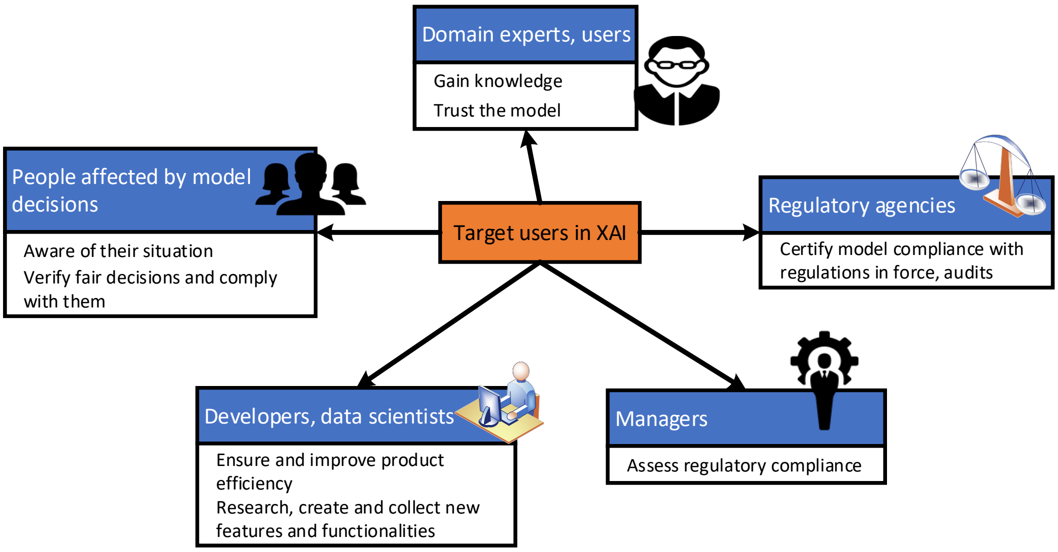

Domain experts, users
Gain knowledge
Trust the model
People affected by model decisions
Aware of their situation
Verify fair decisions and comply with them
Target users in XAI
Regulatory agencies
Certify model compliance with regulations in force, audits
Developers, data scientists
Ensure and improve product efficiency
Research, create and collect new features and functionalities
Managers
Assess regulatory compliance

Figure 2.5: Different purposes of XAI models sought by different audience profiles

developers analyse and erase irrelevant patterns to renovate the dataset.

- **Data Collection**: XAI would give an insight into ML models and a good understanding of the value of the feature in the training set. Thus, the data scientist can evaluate the importance of features and adjust the process of data collecting.

- **Human Decision-making**: In critical sectors where decisions are so important and sensitive, they must be made by humans. XAI can serve as a tool to support humans by providing its predictions with reasonable explanations.

- **Trust Building**: This is arguably the ultimate aim of XAI [75] [76]. By getting an insight into the black-box model, people can verify basic facts or identify errors to avoid. Therefore, XAI builds trust between ML models and humans.

In IDS, identifying malicious behaviour is only the first step because understanding such a decision is crucial for a solution. An insight into the decision helps administrators identify the part of the network, the part of features and the security policies compromised by attackers [77]. With the information provided by XAI, the IDS operator can give the correct actions, whether it is to debug the IDS model or apply new security policies to prevent the same attacks in the future. Considering the benefits of XAI discussed above, the need for XAI in IDS is of utmost need.

2.4.3.2 Taxonomy of XAI

Methods for XAI can be classified into different groups based on some criteria:

- Intrinsic or post-hoc: intrinsic method explain ML models' predictions by restricting the complexity of these models. Meanwhile, post-hoc methods explain the model's results after training by computing their inputs and outputs. The main difference between these two approaches is that most post-hoc approaches can analyze all different ML models. In contrast, the intrinsic approach can only be applied to some specific models whose structures are simple enough to allow intrinsic explainable methods [78].
- Model-specific or model-agnostic: each model-specific method can only be used on a specific model, while model-agnostic methods can be used on many models. When a specific type of explanation is expected, a model-specific approach will limit the choices of black-box models because each model-specific method only provides a type of explanation [69]. Model-specific methods are intrinsic methods, and most post-hoc methods are also model-agnostic methods.
- Local explanation or global explanation: while the local explanation methods focus on a single input data instance and utilize different data features to generate explanations, the global explanation methods work on a subset of the whole data instances to summarize the global behaviour of the model [78].

In the scope of this thesis, the main focus will be on post-hoc techniques, which are widely used to explain DNN models [73]. Nevertheless, first, intrinsically interpretable models will be discussed for comparison.

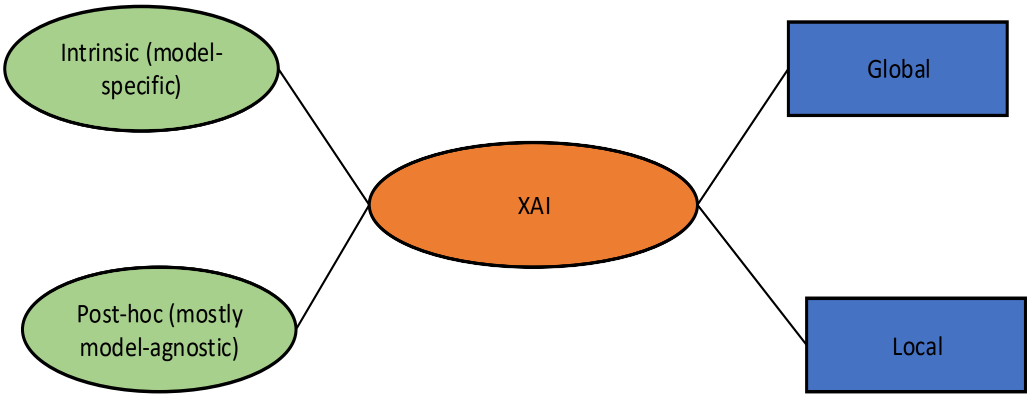


Figure 2.6: Classification of XAI methods

2.4.3.3 Interpretable models

As discussed above, there are ML models that can be intrinsically interpreted due to their simple structure. This allows researchers and developers to impose constraint directly on the ML model to explain its decision based on the internal functioning. A notable example of this approach is proposed in [79]. The authors introduce a generative model named Bayesian Rule Lists based on a decision tree in order to produce interpretable medical scoring systems. The experiment results show that the proposed model is concise and convincing and suggests applying similar models in other fields.

Xu et al. [80] proposed an attention-based approach for a model that can learn to describe images. In the experiments using three benchmark datasets, the method has good performance and be able to effectively explain the results to users through visualization [80]. The authors also suggest visual attention for future work. Another work in [81] introduces Supersparse Linear Integer Model (SLIM) for creating data-driven medical scoring systems. Due to the high level of sparsity and small integer coefficients, the model can interpret the results with qualitative understanding [69]. Other interpretable ML models include linear/logistic regression, general additive models, general linear models, decision trees, k-nearest neighbours, and rule-based learners listed in [73, 74].

Although this approach provides intuitive explanations that are easy to understand, it can only be applied to models that do not perform very well with a large scale of data. Therefore, the trade-off between explainability and performance hinders this approach's adoption [8]. As discussed previously, deep learning can achieve good performance and flexibility that outperforms the traditional machine learning in data with high scale [68]. However, DL's network that consists of multiple layers are considered as black-box due to the difficulties in explaining its functioning. Therefore, post-hoc approach is utilized in this case to interpret the complex black-box models. This approach can be considered reverse engineering that explains predictions without any modification or knowledge about the model's internal functioning [69].

2.4.3.4 Post-hoc Methods

When it comes to ML models that cannot be interpreted intrinsically due to their sophisticated structures, post-hoc techniques are applied to explain the model after it gives predictions. Most post-hoc methods are model-agnostic which can be applied to any ML model. Such an approach has observed many pieces of research recently due to its desirable advantages as followings [82]:

- Model flexibility: the interpretation techniques are not tied to a specific type of ML model. It is up to developers to choose the most suitable post-hoc methods without changing the black-box model or compromising its high performance.
- Explanation flexibility: the form of explanations is not limited. The post-hoc approach introduces a variety of techniques to generate explanations in different formats. Therefore, the best techniques can be chosen to explain the black-box models depending on the type of audience of XAI's. For example, visualization explanation would be adequate to the lay audience, while feature importance may be more suitable for data scientists.
- Low or no cost to switch: switching the underlying model for a new one is convenient without any modification to the presentation of the explanations. This

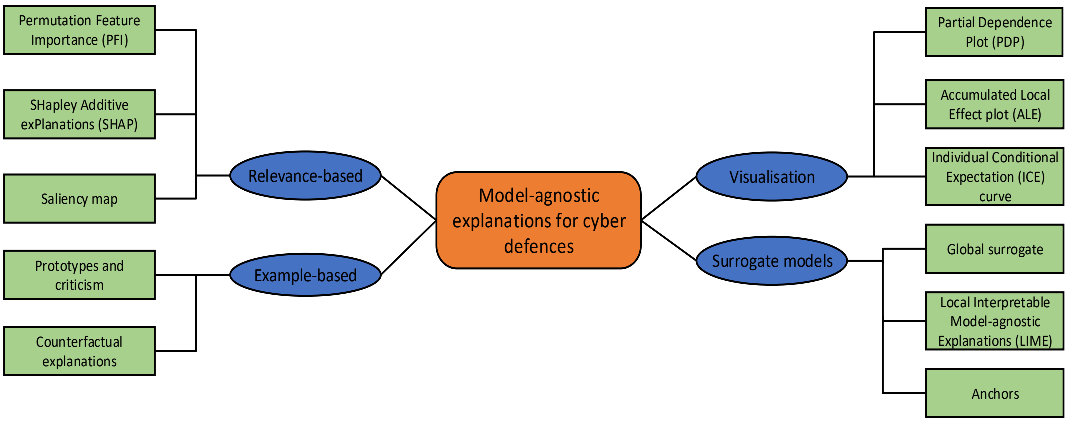

Permutation Feature Importance (PFI)
SHapley Additive exPlanations (SHAP)
Saliency map
Prototypes and criticism
Counterfactual explanations
Relevance-based
Example-based
Model-agnostic explanations for cyber defences
Visualisation
Surrogate models
Partial Dependence Plot (PDP)
Accumulated Local Effect plot (ALE)
Individual Conditional Expectation (ICE) curve
Global surrogate
Local Interpretable Model-agnostic Explanations (LIME)
Anchors

Figure 2.7: Classification of model-agnostic methods

> is a significant advantage over intrinsically interpretable models. Post-hoc methods allow developers to choose the best performing ML model as the underlying model, enhancing accuracy.

Table 2.1 lists all the model-agnostic methods that will be discussed. The advantages/disadvantages of each method and their applications are summarized, the table presents useful knowledge about the state-of-the-art techniques.

Visual explanation is the easiest explanation to understand. This technique creates a visualisation of the model's behaviour from its set of inputs and outputs. Visualisations are the most natural way to demonstrate complex interactions within input features or the effect of each feature on the model's prediction to users who might not have expert knowledge about AI techniques such as domain experts and managers. Visualisation-based as a model-agnostic method is a complex task; therefore, it is usually coupled with feature relevance explanations techniques to improve the understanding and provide comprehensive information to the recipient of the XAI model's results [73]. A list of representative works using this technique can be found in [104, 105]. In the scope of this thesis, some notable techniques of this type will be discussed, including **Partial Dependence Plot** (PDP), **Accumulated Local Effect** (ALE) plot and **Individual Conditional Expectation** (ICE) curves .**Partial Dependence Plot** (PDP) visualise the average marginal effect on global level of a subset of features on the model's

Table 2.1: Comparison of model-agnostic methods

Model-agnostic methods	Advantages	Disadvantages	References
PDP	- intuitive - causal interpretation	- assumption of independence - hidden heterogenous effects - possibly choose only two features	[83–85]
ALE plot	- works with correlated features - faster to compute (than PDP)	- complex implementation - are not accompanied by ICE curves	[86]
ICE curve	- intuitive - uncover heterogenous relationships	- possibly display only one feature - assumption of independence - plot can be overcrowded	[87, 88]
Global Surrogate	- flexible - advantages of the model chosen as surrogate	- explanations would not fit all data instances - disadvantages of the model chosen as surrogate	[89–91]
Local Surrogate (LIME)	- flexible - works for tabular data, text and images	- undefined neighbourhood when applied with tabular data - assumption of independence - explanations can be instable - can be manipulated to hide biases	[75, 92–94]
Anchors	- easy to understand - works with complex predictions in an instance's neighbourhood	- requires a highly configurable setup - conflicting anchors - complex output spaces - realistic perturbation distributions	[95]
PFI	- highly compressed, global insight - consider all interactions with other features	- assumption of independence - only works with labelled data - results vary greatly when the permutation is repeated.	[96, 97]
SHAP	- effects are distributed fairly - based on a solid theory - allow contrastive explanations	- computationally expensive - assumption of independence.	[98]
Saliency map	- intuitive	- fragile - unreliable	[99]
Counterfactual explanations	- easy to understand, facilitate human reasoning	- multiple counterfactual explanations may contradict each other	[100]
Prototype and criticism	- provide meaningful insights	- may be misleading due to irrelevant features	[101–103]

predictions with all other features fixed [106]. Each row of data is considered and predict the outcome by the fitted ML model. Then, the value of the features of interest is altered repeatedly to make a series of predictions.

After iterating through all data instances, the relationship between the output and the features is to visualize. This technique has two major drawbacks: assumption of independence and hidden heterogeneous effect [74]. The term assumption of independence means that the feature of interest is not correlated with other features, which is a false assumption. While the hidden heterogeneous effect may occur since the PDP only visualize the average marginal effect. Each data point can have a positive or negative association with the model's prediction; thus, only presenting the average effects of all points would hide such relationships [74].

A solution to prevent the assumption of independence is the **Accumulated Local Effect** (ALE) plot proposed by Apley et al. [86]. The ALE plot is designed to perform well with highly correlated input features; therefore, there is no need to assume that the features are independent. One more advantage of the ALE plot over PDP is that it requires less computational resource. Goldstein et al. [87] introduce a method called **Individual Conditional Expectation** curves to uncover heterogeneous relationship. ICE plots show one line per data instance, representing the relationship between a feature and the fitted model's predictions. Therefore, comparing ICE curves with PDP would produce interesting insights into ML models as ICE curves of each instance might be significantly different from the average of all instances (PDP) [107]. ICE plots also have some limitations. Firstly, the ICE plot is only useful for one feature since drawing two features will create overlaying surfaces [74]. Secondly, ICE curves suffer from the assumption of independence, just like PDP. Lastly, the plot might get overcrowded, hence become impossible to analyze.

Surrogate models are interpretable models that mimic the behaviour of the black-box models. This post-hoc method is especially flexible due to the free choice of surrogate models. However, the generated explanations would not fit all records in the dataset, and there is no valid theory that supports the surrogate model's representation of the complex model [69]. Moreover, this method suffers from the drawbacks of the interpretable model chosen as the surrogate model. In [75], the author proposed **Local Interpretable Model-agnostic Explanations** (LIME) which derives from the concept of local surrogate models. LIME's key concept is fitting an interpretable model around a specific instance to visualize significant features of that data instance. The interpretable model can be of any type as long as it is a good approximation of the ML model predictions locally [75]. Ribeiro (2016) suggested using LASSO as it performs well comparing to other linear models [75]. LIME is flexible because it can perform well with tabular data, text and images. However, the correct definition of the neighbourhood is questionable when using LIME with tabular data. Because improper kernel settings can lead to non-sense explanations [74]. In LIME, data points are sampled from a Gaussian distribution, which means LIME assumes the independence of features. Consequently, invalid data points may be generated and learned by local explanation models.

In [108] the authors show the instability of the explanations produced by LIME. Moreover, Slack et al. (2019) describe how LIME explanations can be manipulated to hide biases in the dataset [109]. Overall, though LIME is promising, it is in the development phase and needs to be improved a lot before being safely applied. There have been several pieces of research working on fixing LIME's issues or analyze its properties [92–94]. In [95], the creator of LIME introduces an extension method of LIME using high-precision rules class called **Anchors**. Anchors' explanation is intuitive and easy to understand; however, it requires a highly configurable setup like most other perturbation-based explainers. Ribeiro et al. (2018) also discuss Anchors' limitations, including overly specific anchors, conflicting anchors, complex output spaces, and realistic perturbation distributions [95].

Relevance-based methods explain the model by ranking the most relevant input features that impact the model's prediction. **Permutation Feature Importance** (PFI) measures the increase in prediction error of a fitted model after permuting the feature's values. This technique shows the importance of each feature by breaking the relationship between the feature and the desired output. Therefore, the model's error would increase if the feature of interest is considered as being important by the model. PFI was first introduced in 2001 by Breiman in [96]. In 2018, Fisher et al. proposed *Model Class Reliance* which is a model-agnostic version of the PFI [97]. PFI provides global insight into the model's behaviour and automatically considers all interactions with other features. PFI requires shuffling the features, which adds randomness to the computation and makes results vary greatly after repeated training. Moreover, PFI suffers from the assumption of independence.

Lundberg et al. [98] propose **SHapley Additive exPlanations** (SHAP) which is based on the game theoretically optimal Shapley Values in [110]. SHAP connects LIME and Shapley values and has several advantages over LIME [74]. The behaviour of the ML model is assumed to be linear locally in LIME, yet the effects are distributed fairly in SHAP. Moreover, SHAP allows contrastive explanations by comparing a prediction to a subset or a single data instance. However, SHAP has to use all the features, hence being computationally expensive. Assumption of independence is also a big problem with SHAP, like many other permutation-based interpretation methods. Another method named saliency maps (or pixel attribution map) [99] is a type of both relevance-based methods and visualization explanations as a pixel of an input image can also be considered as a feature. These approaches generate intuitive explanations as they highlight the most relevant pixels on the final classification. Importance scores of individual pixels are computed using occlusion techniques, or calculations with gradients [69]. However, the saliency map would be fragile [111], and very highly unreliable [112].

Unlike other model-agnostic methods, example-based explanation approaches generate explanations from particular instances of the dataset instead of creating the summaries of the features [69]. Example-based explanation methods have two main techniques, including counterfactual explanations and prototype and criticisms. Wachter et al. [100] introduce counterfactual explanations as a novel model-agnostic XAI method. The technique explains the ML model by describing the change needed in an instance to change or flip the prediction. Explanations generated by this technique is straightforward for human to understand because human usually asks why a specific decision was made instead of other decisions [113]. However, each instance can have multiple counterfactual explanations, and they may contradict each other [74].

Prototypes are representative data instances of the dataset [101,102] and criticisms are data points that are not well represented by the prototypes [103]. Together with criticism, prototypes can provide meaningful insights into the ML model. Kim et al. [103] develop MMD-critic that selects prototypes and criticism for a dataset to aid human understanding and reasoning. In addition, this technique requires a meaningful data-processing module to select only relevant features because prototypes and criticisms are generated by taking all the features, which may be misleading due to irrelevant features [74].

In the scope of this thesis, the two following classes of techniques will be focused on:

- **Feature importance explanations**: the technique is a type of relevance-based method that explains the model's decisions by calculating an importance score for each feature. A comparison among different features' scores would reveal the importance of each feature, which is granted by the model when making decisions [73]. A popular XAI post-hoc method using feature relevance techniques mentioned above - SHAP proposed by Lundberg et al. in [98]. The proposed method combines LIME with Shapley value to generate local explanations for the model and demonstrate the relationship between values of input features and the model's decisions.
- **Visual explanations**: this technique utilise the input features and model's prediction to visualise the model's prediction usually by plotting graph. Visualisation-based method is effective in explaining a black-box model to various recipients who are not expert in AI techniques. Visualisation-based as a model-agnostic method is a complex task; therefore, it is usually coupled with feature relevance explanations techniques to improve the understanding and provide comprehensive information to the recipient of the XAI model's results [73].

Based on the concepts discussed above, tools on XAI were developed such as DeepVis Toolbox, TreeInterpreter, Keras-vis, Microsoft InterpretML, MindsDB, SHAP, Tensorboard WhatIf, Tensorflow's Lucid, Tensorflow's Cleverhans and many others. Most of these tools are model-agnostic methods, and a few are model-specific. For instance, DeepVis, kerasvis, and Lucid are for a neural network's explainability, and TreeInterpreter is for a tree-based model's explainability [72]. Each of the proposed approaches has similar concepts at a high level, such as relevance-based, Shapely values, partial dependence plot, surrogate models, counterfactual, prototype, and criticism.

2.5 Internet of Things Explainable AI for Cyber Security

In recent years, the term Internet of Things (IoT) is gaining popularity in wireless telecommunications. Its first definition was given by Ashton in the title of a presentation at Procter & Gamble (P&G) in 1999 [114], and the definition directly referred to RFID. Since then, the scope of IoTs has been developed and extended beyond the scope of RFID technologies [115]. According to International Telecommunication Union (ITU), IoT is a global infrastructure for the information society which enables services by interconnecting things based on existing and evolving information and communication technologies [116]. At the same time, alternative definitions have been proposed to emphasize different subjects such as connected things in IoT, Internet-related aspects of IoT, semantic challenges in the IoT and many others [115].

Generally, the term IoT refers to the network of physical devices, objects, vehicles, buildings embedded with sensors, software, and other communication technologies to collect, store, analyse and exchange data with other devices and systems. Evolving technologies, which build IoT, are characterized by the development of low-powered, low-cost processors, wireless networking, artificial intelligence, and mobile computing [117]. Together, these technologies make communication between people, processes, and things become much more accessible, improving the efficiency and quality of human life. Therefore, technologies related to IoT has been advanced rapidly, making it the fastest-growing technology in computing [1]. By the year 2025, IoT and related applications are estimated to have an economic impact of \$3.9 trillion to \$11.1 trillion per year [118].

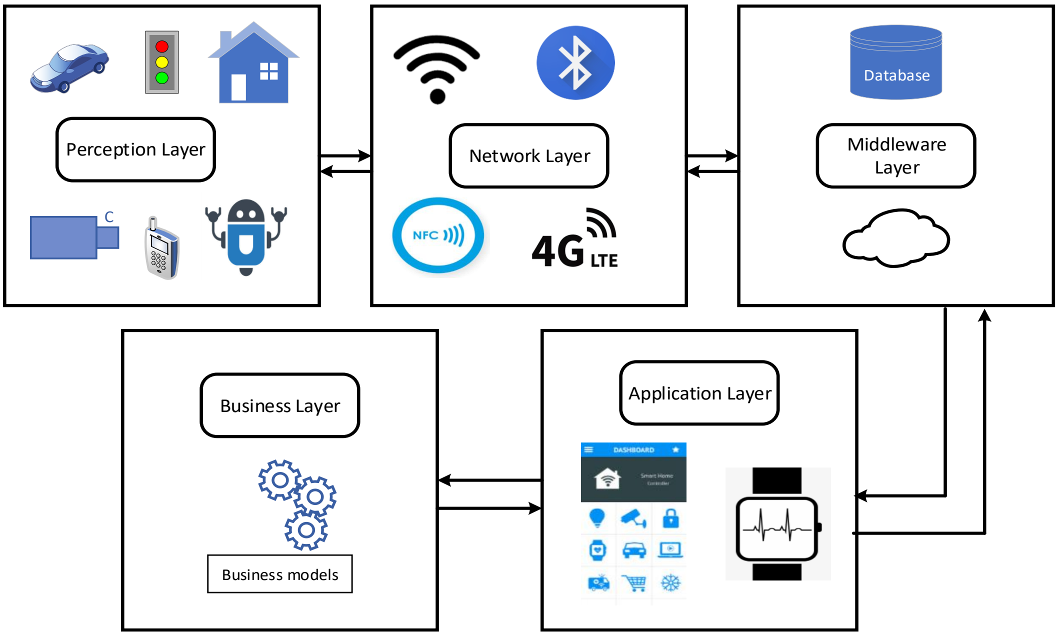


Figure 2.8: Five-layer architecture of IoT

The domain of IoT typically consists of a wide range of advanced technologies; thus, there is no single reference architecture that can represent all possible implementation of IoT networks [119]. However, for research purposes, the five-layer IoT architecture is utilized to focus on the finest aspects of IoT which can fulfill the requirements of security and privacy [120]. Figure 2.8 shows the five-layer architecture of IoT. The perception layer is the sensor layer that collects information through the sensors attached to physical objects such as cars, robots, surveillance cameras, phones and many others. The network layer connects devices and servers and transmits collected data from the perception layer using Wifi, Bluetooth, Near-Field-Communication (NFC) and other methods. Middleware layer process and analyze transportation data. It purifies the data and only extracts useful information [120]. Cloud computing and big data processing modules are examples of technologies used in this layer [119]. Application layer refers to IoT applications that provide services to users such as smart home, smart health and many others. Lastly, the business layer is responsible for controlling IoT applications, business models and user privacy [121].

IoT data were used to sent to cloud for processing, then IoT devices receive signals from cloud for further actions. However, this old approach was replaced by the emergence of fog computing (also known as edge computing) due to many advantages that the fog can offer. Fog nodes are devices that extend the cloud to be closer to the devices that produce and act on IoT data. Fog nodes can be any devices with computing, storage and network connectivity such as industrial controllers, switches routers and surveillance cameras [122]. Generally, fog was developed to address two major problems in IoT network including high network latency and sensitive data leakage. IoT applications are written for fog nodes at the network edge, allowing fog nodes to ingest IoT data from multiple devices. Then, different types of IoT data is directed to a proper place for analysis based on the level of time sensitive; the places can be fog nodes, fog aggregation nodes or cloud [122]. Therefore, fog nodes offer advantage for IoT network in deploying distributed and parallel security services due to its ability to offload heavy computations from IoT devices [123].

In recent years, several pieces of research [44,124] has focused on utilising fog computing to design a distributed IDS in IoT system. In regards to the advantages of this approach, we recommend utilising fog computing to design a distributed explainable AI-based IDS. Additionally, different explanations generated from the IDS will be directed to the optimal place for analysis and inform multiple audiences. Figure 2.9 illustrates the architecture of model for anomaly-based IDS in IoT network. First of all, telemetry data from various IoT devices are sent to fog nodes for pre-processing. Then, an ML-based decision engine classifies the activity as normal or attack. Normal activities are allowed to happen. In contrast, abnormal traffic is terminated and sent to the XAI model for processing. The model generates different types of explanation that is directed to inform audiences or being stored in a cloud server for further research.

2.5.1 Cyber threats in IoTs network

IoT has appeared in all aspects of society, including industry, healthcare, homes, sport, peer-to-peer networks [125], entertainment, and others [126]. It is gradually replacing many conventional computer systems in those fields. However, in comparison with con-

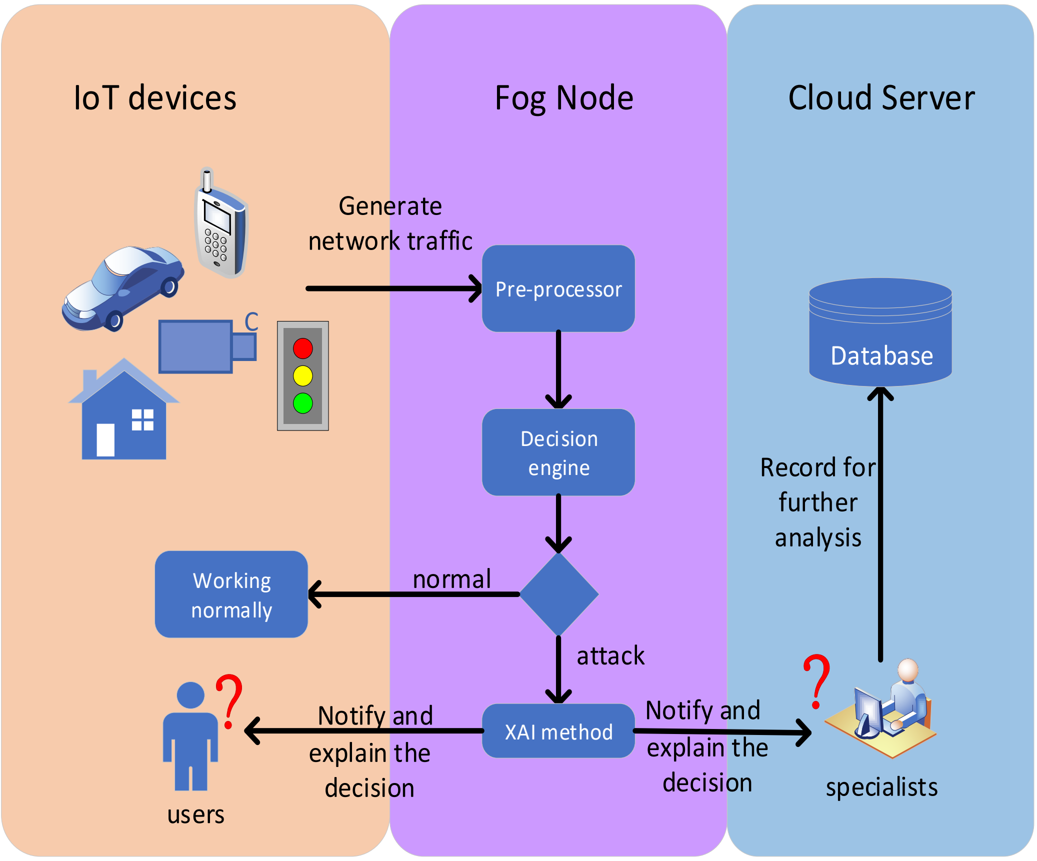


Figure 2.9: Architecture of an explainable AI-based IDS in IoT network

ventional computing systems, this engagement has introduced new security challenges for several reasons [1]:

- **Complex and diverse environments**: IoT is diverse as it is connected with a wide variety of devices, platforms, communication means and protocols. The diversity enhances the usability and convenience of IoT technologies yet at the cost of numerous potential targets and attack vectors.
- **Undefined boundary**: The IoT system does not have a well-defined boundary since it changes very often due to the mobility of users and devices. Due to the undefined boundary, it is not easy to design effective security mechanisms for the IoT system. Moreover, the IoT systems would suffer from an extensive attack surface.
- **Connection between virtual and physical systems**: In unattended working environments, IoT devices would be a potential target since the attacker can access them physically. Moreover, IoT devices can function on the received data, which optimises the connection between virtual and physical systems yet allows the attacker to convert the potential physical consequences quickly.
- **Limited energy and computational resources on devices**: Most IoT devices have limited energy and computational resources, making it hard to implement decentralised IDS and advanced security techniques on physical objects. For example, IDS that utilises deep neural networks require GPU to perform well in real-time, which most IoT devices cannot afford.

Regarding the reasons discussed above, IoT networks are attractive targets for cyber attacks. Many attack techniques are utilised to exploit the security issues at different layers of the IoT environment, including sensing, network, middleware, application and business layer. According to [127], 41% of attacks exploit IoT devices' vulnerabilities; then, victim devices can be utilized to launch a large-scale attack. For example, a Mirai-based attack compromised a French cloud computing company in 2016 and became the most significant distributed denial of service attack (DDoS) recorded at that time

[2]. Attackers were reported to instrument zombified IoT devices and use them as a pivot point to launch a DDoS attack on the French web host. Researchers then blame default, and weak security configurations [1]. This example has testified to the secure authentication mechanisms and traffic classification techniques. Thus, to prevent cyber threats in IoT systems, new defence mechanisms should be developed. IDS is considered as the primary method to attain these requirements [128].

2.6 Related Studies

2.6.1 ML-based IDS

In recent years, the use of ML algorithms and techniques in holistic IDS has become common. Researchers produced many pieces of research related to the application of ML in IDS. Rahul et al. [129] used the KDDCup-99 dataset to compare the performance of several classical ML algorithms and Deep Neural Networks (DNN) models, which have from 1 to 5 layers. After running 100 epochs, the DNN model with three layers outperformed all other DNN models and classical ML algorithms. Results, shown in Table 2.2, demonstrate the efficiency of deep learning models in IDS.

Different from other detection methods, DL-based IDS must consider handling overfitting and model optimization. Overfitting means that the model works well on training data but it has poor performance on unknown records; therefore, it is ineffective in real life. Model optimization aims to minimize a loss function to improve the effectiveness of the model by applying optimizer such as Stochastic gradient descent (SGD), Adam and others. In regards to those characteristics and a few others including accuracy rate and performance comparison, table 2.3 summarises the advantages and disadvantages of existing literature in DL-based IDS.

KDDCup-99 was used in [130] to train CNN (convolutional neural network), CNN-RNN, CNN-LSTM and CNN-GRU. The experiment was run up to 1000 epochs with a learning rate in the range [0.01-0.05], and results were recorded. The complex network

Table 2.2: Performance comparison

Algorithm	Accuracy	Precision	Recall	f1-score
DNN (1 layers)	0.929	0.998	0.915	0.954
DNN (2 layers)	0.929	0.998	0.914	0.954
DNN (3 layers)	0.930	0.997	0.915	0.955
DNN (4 layers)	0.929	0.999	0.913	0.954
DNN (5 layers)	0.927	0.998	0.911	0.953
Ada Boost	0.925	0.995	0.911	0.951
Decision Tree	0.928	0.999	0.912	0.953
K-Nearest Neighbour	0.929	0.998	0.913	0.954
Linear Regression	0.848	0.989	0.821	0.897
Naive Bayes	0.929	0.988	0.923	0.955
Random Forest	0.927	0.999	0.910	0.953
SVM* (linear)	0.811	0.994	0.770	0.868
SVM* (rbf)	0.811	0.992	0.772	0.868

structure suffered from overfitting and were outperformed by simple network structures. Results showed that CNN 1 layer obtains the best accuracy (0.999), precision (0.999), recall (0.999) and F-score (0.999). This paper applied CNN for IDS by modelling the network traffic events as time-series of TCP/IP packets. From the experiment results, the authors claimed that modelling network activities as series of TCP/IP packets are an efficient method to train DL models such as CNN, RNN, LSTM, or GRU [130].

In [131] and [132], the authors also used KDDCup-99 to train LSTM-RNN for anomaly-based IDS, and both papers obtained the best detection rate of 98.95%. The authors in [131] changed the learning rate and the number of hidden nodes to test the results, while different types of optimizers were used in [132] to compare the performance. In [133], Kim and Kim (2015) applied RNN with hessian-free optimization and obtained a reasonable detection rate of 95.37% and a low false alarm rate of 2.1%.

Xu et al. [134] combined multiple techniques to build an IDS using KDDCUP-99 and NSL-KDD datasets. In the proposed model, GRUs are utilized to build the main memory unit, and MLP was used to classify intrusions. The results showed the detection rate of 99.42% and 99.31% on KDDCup-99 and NSL-KDD, respectively [134]. However, the authors did not claim which specific dataset the model was tested on, neither KDDTrain+, KDDTest+, nor KDDTest-21. Yin et al. (2017) proposed an RNN model and conducted testing by changing the number of hidden neurons and learning rate to find the optimal hyperparameters [135]. The RNN model and other models using conventional ML algorithms were tested on KDDTest+ and KDDTest-21. Results showed that RNN outperformed other methods in both binary and multiclass classification. In the binary classification, RNN achieved an accuracy of 83.28% and 68.55% in KDDTest+ and KDDTest-21, respectively. In the multi-class classification, the accuracy were 81.29% and 64.67% in KDDTest+ and KDDTest-21, respectively.

Andalib and Vakili [136] proposed an IDS combining three different types of ML techniques (GRU-RNN, CNN and RF) using the NSL-KDD dataset for training and testing. Results showed that the proposed model can achieve good accuracy on KDDTest+ and KDDTest-21, which were 87.28% and 76.61%, respectively. Moreover, Andalib and Vakili (2020) claimed that the ML model has a short training time and only needs low computational resources [136]. Chaibi et al. (2020) proposed an architecture implemented as two methodologies, then trained and tested the model on the NSL-KDD dataset for five class attack categories [137]. Results showed that RNN outperformed ANN and ANN outperformed other ML classifiers. However, the paper has the identical drawback with the paper in [134] where the authors did not provide performance on the challenging testing dataset - KDDTest-21.

Gwon et al. (2019) proposed a LSTM with feature embedding to build an IDS and utilized UNSW-NB15 for training and testing. The experiment assumed that the data instances follow a timely order, which means that the dataset captures temporal dependence for intrusion detection [138]. LSTM models with feature embedding can obtain the accuracy of 99.72% for binary classification and 86.98% for multiclass classification, which outperformed MLP and other conventional ML algorithms [138]. The authors contributed excellent performance to embedding techniques. Due to the distinguishable information of features in the UNSW-NB15 dataset, feature embedding is an efficient way to capture such information to design an effective model [138]. Roy and Cheung [139] built a Bi-Directional LSTM-RNN for anomaly-based IDS and obtained the accuracy rate of 95.71% with a false alarm rate of 0%. However, the model was only trained and tested on a small subset of the UNSW-NB15 dataset. Tama and Rhee [140]

Table 2.3: Deep learning-based IDS

Reference	Techniques	Dataset	Advantages	Disadvantages
[129]	DNN(1-5 layers), classic ML	KDD-CUP99	- good comparison between multiple techniques	- no model optimization - outdated and unbalanced dataset
[130]	CNN with CNN-LSTM with CNN-GRU	KDD-CUP99	- experiment with different configurations	- overfitting - no model optimization - outdated and unbalanced dataset
[131]	LSTM-RNN	KDD-CUP99	- high detection rate - experiment with different configurations	- outdated and unbalanced dataset
[132]	LSTM-RNN	KDD-CUP99	- high detection rate - experiment with different configurations	- overfitting - outdated and unbalanced dataset
[133]	RNN with Hessian-free optimization	KDD-CUP99	- low false alarm rate	- overfitting - outdated and unbalanced dataset
[134]	GRU with MLP	KDD-CUP99 NLS-KDD	- comparison between two datasets - good accuracy	- do not claim which specific dataset the model has been tested on - overfitting - no optimization - outdated dataset
[135]	RNN	NLS-KDD	- comparison between RNN and classic ML	- overfitting - no optimization - outdated dataset
[136]	GRU-RNN with CNN with RF	NLS-KDD	- good accuracy - short training time - low computational resources - experiment on different configurations	- outdated dataset
[137]	RNN, ANN, classic ML	NLS-KDD	- comparison between multiple techniques	- do not provide testing results on challenging dataset KDDTest-21 - outdated dataset - overfitting
[138]	LSTM with feature embedding, MLP, ML	UNSW-NB15	- comparison between multiple techniques - feature embedding	- no optimization
[139]	Bi-LSTM	UNSW-NB15	- feature selection (only consider 5 features)	- only use a small subset of UNSW-NB15 - overfitting - no optimization
[140]	DNN	UNSW-NB15, CIDDS, GPRS	- comparison between different datasets	- no optimization - lack comparison between multiple techniques
[141]	LSTM, ML	CIDDS	- good comparison between multiple techniques	- overfitting

proposed an IDS built upon DNN model for IoT network using new datasets including UNSW-NB15, CIDDS-001 and GPRS. The model performed perfectly on CIDDS-001 in which it achieves about 100% accuracy, precision and recall. However, the paper lacks a performance comparison between DNN and other ML algorithms. The research in [141] used the CIDDS dataset to train and test LSTM model. The performance was compared between different ML algorithms, and LSTM outperformed all other methods with an accuracy of 84.83%.

2.6.2 DL-based IDS for IoT

The Bot-IoT dataset [142] is a IoT benchmark dataset suffering from an imbalanced data problem in which it has a small amount of benign data and a large amount of attack data. In regards to the solutions for balancing this dataset and other characteristics, table 2.4 summarises the advantages and disadvantages of recent literature in DL-based IDS for IoT.

Ferreg and Maglaras [143] proposed a DeepCoin framework built upon blockchain and deep learning. The proposed model utilized RNN for IDS and evaluated the performance on three datasets including the Bot-IoT dataset and two others. The authors balanced the dataset before using it for training the model. Although model optimization was not used, the IDS achieves a good accuracy of 98.20% on the Bot-IoT dataset. Aldhaheri et al. [144] develop Deep Learning and Dendritic Cell Algorithm (DeepDCA) to design an IDS. The proposed model extracts features from Bot-IoT then uses it for training and evaluating. The authors use a balanced version of the dataset but do not provide information about extracting the dataset.

The creators of the Bot-IoT dataset also use the created dataset to test RNN and LSTM [142]. Before training the model, they pre-process the dataset by calculating the correlation coefficient among the features and use normalization to scale features' value within the range [0, 1]. To solve the imbalanced data problem in Bot-IoT dataset, authors in [145] and [146] use Synthetic Minority Oversampling Technique (SMOTE) before training the proposed IDS built upon ANN and Temporal CNN, respectively.

Table 2.4: Deep learning-based IDS for IoT

Reference	Techniques	Dataset	Advantage	Disadvantage
[143]	RNN	Bot-IoT	- balanced dataset - high accuracy	- no model optimization
[144]	DeepDCA	Bot-IoT	- high accuracy - loss function as model optimizer	- no information about how dataset is balanced
[142]	RNN, LSTM	Bot-IoT	- compute correlation coefficient among features	- overfitting - unbalanced dataset - no model optimization
[145]	ANN	Bot_IoT	SMOTE to balance dataset	- overfitting - no model optimization
[146]	Temporal CNN	Bot-IoT	SMOTE to balance dataset	- online mode was not implemented for real time detection
[147]	feed-forward neural network	Bot-IoT	balanced dataset	- model only works in batch mode - poor encoding technique for port columns
[148]	GBM, RF, NN	ToN_IoT Aposemat IoT-23	- novel approach of cross-training - comparison between IoT datasets - high accuracy	- default parameter settings for training NN model
[149]	GBM, RF, NB, DNN	ToN_IoT network	- novel testbed architecture - high accuracy	- only a small part of the dataset was used - used IP addresses and ports to build models

The TCNN model shows promising results with low training time; however, it was not implemented in online mode for real time detection.

Ge et al. [147] propose a feed-forward neural networks model for binary and multi-class classification of intrusions in IoT devices. They use a balanced version of the Bot-IoT dataset and Adam as the model optimizer. The performance is good; however, the model only works in batch mode. Moreover, the author use a list of well-known ports for encoding port columns. Booij et al. [148] evaluated multiple classifiers on two IoT datasets including ToN_IoT and Aposemat IoT-23. They conducted a novel approach of cross-training by using data fusion on the data level and observed that training on one dataset and testing with the other provide bad results. Therefore, the inclusion of configurations is crucial. In the experiment, they used default parameter settings for training neural network; therefore, the neural network is outperformed by Gradient Boosting Machine (GBM) and Random Forest (RF).

Moustafa [149] presented a novel testbed architecture that was used to collect heterogeneous data sources from IoT/IIOT devices, Windows and Linux-based operating systems, as well as network traffic. 4 different classifier was built and evaluated by using a small part of the ToN_IoT network dataset. Moustafa used the IP addresses and ports when building models to collect benchmark outputs for comparison purposes. It is recommended to exclude IP addresses and ports in the data features.

2.6.3 Explainable AI-based IDS

To the best of our knowledge, there are only a few work-related to XAI in IDS. Most of the XAI's works focus on fields such as computer vision and natural language processing. An attempt of designing an XAI model in IDS is proposed in [77]. The authors apply a decision tree to build an ML model, then train and evaluate the performance on the KDD benchmark dataset. Because the decision tree is an interpretable model, it can explain the decisions using feature engineering, and the rule-based model [77]. Although the decision tree model outperforms logistic regression and support vector machine in predicting the classes between malicious and normal behaviours, its performance is not compared with the deep learning model.

Islam et al. (2019) propose a method to gather and utilize domain knowledge to automate the defence response and improve the explainability of the IDS model. In the experiment, domain knowledge (i.e., CIA principles) is instilled into the ML models, and the CIDS2017 dataset is utilized [150]. Results from the experiment show an increase in generalizability and explainability of the models, which builds trust in the IDS model and opens the door to adapt to big data from numerous IoT devices [150].

Marino et al. (2018) produced explanations for incorrect samples by using an adversarial approach [151]. It finds the minimum modification to correctly label the incorrect records then visualize the most critical features to explain the model's wrong decision. The authors experiment on the NSL-KDD99 dataset. Some advantages of the approach include (1) being a model-agnostic method, (2) require no modification to the internal structure of the black-box model and (3) being extendable for further analysis of the model [151].

Le et al. [152] propose a IDS built upon an RNN and explain the outcomes of the model to generate Software Defined Networking (SDN) flow rules. They use a linear regression model as a local surrogate describes in LEMNA [153], the k most relevant features are chosen to generate network access control policies. The authors use NSL-KDD as a benchmark dataset.

2.7 Research Challenges and Opportunities

2.7.1 Challenges to building an efficient NADS in IoT networks

Despite many pieces of research in NADS showing good performance with benchmarking datasets, it is challenging to build an effective IDS with a high detection rate, scalability, robustness and protection against all attack vectors [1]. This section presents challenges for designing an efficient IDS in IoT networks.

- Data source is a key component of NADS in both the training and testing process. However, it is impossible to construct a dataset that involves all normal and malicious behaviours in a heterogeneous environment of IoT networks. Many existing datasets suffer from missing labels and poor attack diversity. Moreover, most of them collect an incomplete set of features, and network information is captured without including both headers and payloads [14]. In addition, the IDS may only perform well on a limited number of devices because collected data only contains network activities of a few types of IoT devices [1].

- Real-time detection is also a challenge. Collecting and monitoring network traffic in real-time would cause long processing time and high false alarm rate [14], which can degrade the network performance significantly. Therefore, the data-processing module and the detection method must be adopted carefully to mitigate this [14].

- Although the IDS using DL-based techniques shows good performance, apply such techniques in IoT environments is difficult due to the limited computation resources on IoT devices. Moreover, ML and DL-based techniques are computationally expensive, which leads to network latency issues and hence, become impossible to be used in critical sectors such as heal or internet of vehicles (IoVs) [154].

2.7.2 Challenges to achieving XAI

A noticeable challenge in achieving XAI is accessible explanations. Instead of focusing on the demands of different types of audience as in Fig 2.5, most current XAI methods produce results that only make sense to researchers [15]. In addition, some XAI methods produce explanations as feature important vectors, which is inherently low-level explanations. This format would be useful for developers and researchers to debug or design the models. However, other audience types may find such explanation format complex, confusing and useless [15]. Ideally, XAI method should provide accessible explanations for all types of audience, which means they should be easily utilised by society, especially policymakers and the law [73]. This remains a big challenge in the field of XAI.

2.7.3 Research Opportunities

The field of explainable aritificial intelligence (XAI) emerged from the need to understand the predictions and functioning of machine learning (ML) models, and to develop robust XAI models that benefit various types of audience [73]. XAI methods have witnessed many studies and applications across a wide of range of subjects, such as image recognition [155–157] and natural language processing [158]. However, there have been very few studies in intrusion detection systems (IDS). Nowadays, as human life significantly depend on computer systems, IDS is crucial for security of any organisation's system. Additionally, there is growing need for explainable AI-based IDS since the model's recipients require explanations to either verify the decisions or improve the model's performance. Throughout the literature, there are many examples of ML-based IDS (subsection 2.6.1, 2.6.2) and various XAI methods (subsection 2.4.3) with very few addressing explainable AI-based IDS (subsection 2.6.3).

This chapter identifies a gap in existing research that is of utmost importance. The field of XAI is relatively young, and there are limited existing literature addressing the explainable AI-based IDS. There is a lack of studies in creating new XAI methods for IDS and the evaluations of existing XAI approaches for ML-based IDS. This provides an opportunity for research, to develop an understanding of existing XAI methods for IDS and design a novel explainable AI-based IDS. This thesis proposes to fulfill the gap in research through the conduct of detailed evaluation of XAI methods in ML-based IDS and exploring the effectiveness of new specific methods for ML-based IDS.

2.7.3.1 Research Questions and Solutions

This thesis aims to enhance the field of Explainable Artificial Intelligence in Intrusion Detection System in IoT network environment. As the research opportunities have been identified, the research conducted throughout this work will address the following question:

How can we design an explainable Artificial Intelligence-enabled Intrusion Detection System that is effective and adaptable to ever-changing Internet of Things network environment?

Since this is a broad and complex question, we divide it into three following sub-questions.

Sub-question 1: *How robust are existing Explainable Artificial Intelligence methods in interpreting decisions of Machine Learning-enabled Intrusion Detection Systems?*

The survey conducted throughout the Literature Review chapter has addressed this question comprehensively. To be more specific, section 2.4 has reviewed the current state of XAI methods, analysing advantages and disadvantages of each method. Next, section 2.5 proposes a model of XAI framework for Cyber Defence systems in IoT networks, considering the architecture of IoT platform. Existing works on this field have been discussed in section 2.6. Finally, section 2.7 has detailed the research challenges and limitations to developing an explainable AI-based IDS in IoT network.

Sub-question 2: *How can we build an Explainable Artificial Intelligence-enabled Intrusion Detection framework in IoT networks?*

In chapter 3, we will build the decision engine of IDS based on Deep Neural Network with optimised parameters, showing a high performance on contemporary IoT network datasets, including NSL-KDD, UNSW-NB15 and ToN_IoT. Then, the explainable framework is designed to interpret the model's decisions and generated explanations will be evaluated through comparing with the real characteristics of attack classes presented in those datasets.

Sub-question 3: *Which architectures ensure Intrusion Detection System are effective, explainable and adaptable to an Internet of Things network environment?*

In chapter 4, we will utilise the results obtained from the proposed framework to build an effective and adaptable IDS. The generated explanations will optimise the feature selection process, enabling effective training of detecting individual attack class in the datasets. The framework can give both global and local explanations for the model's decisions. Moreover, the separate training of major classes in each dataset will give more advanced explanations, and allow the architecture to be adaptable to ever-changing cyber-attacks in IoT network environment.

2.8 Chapter Conclusion

The literature review has examined the importance, current challenges, and recent works in network security, Intrusion Detection Systems (IDS), Artificial Intelligence (AI) and security issues in the Internet of Things (IoT) networks. Since digital transformation is happening across many industries, the security and privacy of computer systems arise as a big concern. Intrusion Detection System is a prominent method to protect cyberspace due to its convenience and automation without the need for human action. Different types of IDS can fit a wide range of organisations depending on the IDS placement and detection method. Recent literature shows that anomaly-based IDS, which uses machine learning algorithms and deep learning as an underlying detection method, achieves excellent results in preventing unknown attacks. Moreover, the heterogeneous nature of IoT also encourages the utilization of AI techniques in IDS due to AI's capability of analysing and learning attack patterns from a large scale of data. This literature review has surveyed recent works in ML-based IDS, especially DL-based IDS. It is observed that many proposed models suffer from outdated or unbalanced datasets, which can degrade the model's efficiency in real-life implementation.

In recent years, the field of explainable artificial intelligence (XAI) has witnessed a significant development due to user's need for explainability and XAI's notable benefits. Ultimately, XAI allows and improves the application of AI techniques across many industries, especially critical sectors which require the explainability of decisions. Within the field of XAI, this literature review has focuses on examining the state of the art XAI methods and summarise their advantages as well as disadvantages. In the next chapter, an explainable deep learning-based IDS is proposed to address the challenge of explaining the most representative features of cyberattack types.

Chapter 3

An Explainable Deep Learning-enabled Intrusion Detection Framework in IoT networks

3.1 Introduction

Emerging technologies such as the Internet of Things (IoT), Smart Cities, mobile devices and the Internet have fundamentally changed how modern society operates [1]. However, such technologies are increasingly complex and pose unique challenges to secure, in addition to increasingly serious impacts [159]. There are several underlying reasons for this; such systems are comprised of a wide variety of low-cost devices, and generate significant data across multiple networks, protocols and for different use-cases [1]. Due to this proliferation of heterogeneous devices, IoT networks generate

[1] The work in this chapter has produced this paper (**Pham, N**; Moustafa, N; Turnbull, B. An Explainable Deep Learning-enabled Intrusion Detection Framework in IoT networks, Future Generation Computer Systems, (under review), 2021)

high-dimensional and multimodal data, which requires the capability of analysing big data. Artificial Intelligence (AI) is one paradigm increasingly sought after to help solve these issues, as AI technologies, especially Machine Learning (ML) and Deep Learning (DL), have been utilised across industries and achieve excellent performance with large scales of data.

IoT networks have unique properties that require different approaches to defence than those used in traditional corporate environments. To prevent malicious activities and protect IoT networks, multiple security mechanisms have been proposed. Traditional approaches, such as anti-malware, firewalls, user authentication and data encryption, are all well-known, and each of them fits different purposes [12]. However, traditional mechanisms are less effective as they are less flexible and dynamic in the face of the rapid growth of attack techniques [13]. Intrusion Detection Systems (IDSs) are a mature and prominently used cybersecurity detection control that can identify diverse cyber attacks and even zero-day attacks [160]. Due to the application of AI techniques, AI-based IDSs can achieve high performance with benchmark datasets. Deep learning techniques can analyse complex data and learn from previous attack patterns in the dataset to detect zero-day attacks. However, such techniques typically have a high false-positive rate and their decisions are opaque to users [3].

Anomaly-based detection is important to discover unknown attacks and protect information technology systems. Therefore, many ML algorithms have been proposed to design efficient IDSs with high accuracy and low false-positive rates. Moreover, the need to explain the functioning and predictions of ML-based IDS arises as different types of users are benefited from understanding the root cause of intrusion detection. Therefore, the field of explainable artificial intelligence (XAI) has been developed in recent years to address this concern. XAI illuminates the black-box model by providing explanations on their functioning and predictions. One of the key limitations of existing deep learning-based IDS models is the discovery of zero-day attacks and interpretation of how the models successfully discovered them [3, 13]. Current signature-based IDSs rely on known attacks signatures and suffer from high false negative and false positive rate, hence limiting their ability to detect unknown attacks and hindering their practical use [161]. Such systems are known to be efficient, but have a high probability of a motivated attacker being able to bypass identification. Additionally, signature-based IDS platforms are unable to discover new or zero-day attacks - they are limited to signatures for attacks that have been previously analysed and are within the database. Moreover, researchers are encouraged to work on interpreting machine learning models as the need to explain their decisions has been written in regulations [162].

This chapter proposes a novel SPIP (S: SHapley Additive exPlanations, P: Permutation Feature Importance, I: Individual Conditional Expectation, P: Partial Dependence Plot) framework. The proposed framework provides both global and local explanations to serve different purposes. The generated explanations are intuitive and useful for lay users, security experts and researchers. Its ultimate aim is to promote the interpretability and explainability of IDSs, hence enhancing the performance of cyber defence systems.

The major contributions of this chapter are structured as follows:

- We propose the novel SPIP framework that generates global and local explanations to the decisions of IDS regardless of the underlying algorithms. This framework promotes the interpretation and explainability of the IDS, hence building users' trust in the IDS. Moreover, this framework support experts in analyzing the IoT datasets and characteristics of various cyber-attacks, which would enhance the performance of cyber defence systems.
- We employ a set of the most important features extracted by the proposed framework and the original set of input features to train the AI-based IDS and compare the performance. The utilisation of a customized set of input features increases the performance and reduce the training time as well as the total detection time of the IDS.

The remainder of this chapter is organized as follows. Section 3.2 discusses the related works. The methodology of this framework is described in section 3.3, including Deep Neural Network (DNN), Shapley Additive Explanations (SHAP), Permutation Feature

Importance (PFI), Individual Conditional Expectation (ICE) and Partial Dependence Plot (PDP). Section 3.4 proposes SPIP framework. Section 3.5 presents the experiments and the results' discussions. Finally, section 4.5 summarises this work and discusses the future research direction.

3.2 Related Work

Intrusion Detection System (IDS) is a prominent control in cyber defence, and used extensively to effectively detect and prevent cyberattacks. IDS was first introduced by Denning [5] in 1987. There are various approaches to divide IDS into different categories; with network-based IDS and host-based IDS being the two main types, which are categorized based on places of deployment. Whilst network-based IDS detect attacks by capturing and analysing network traffic between devices, host-based IDS monitor processes and system events on the host to identify malicious activities [163]. Moreover, an IDS can be design based on two different detection methods, including signature-based and anomaly-based. Whilst signature-based IDS rely on a database consisting of known attacks' signatures to detect them, anomaly-based IDS identify malicious activities by comparing them to a pre-built baseline of normal behaviours [51]. Hybrid IDS employ both methodologies.

Explainable Artificial Intelligence (XAI) was first introduced in the 1980s [70]. However XAI has seen less notice as AI's growth has focused on predictive performance. In recent years, the need of XAI is increasing as AI models are widely used in critical sectors where explanations are required for any decisions. Morevover, XAI can enhance the use of AI models as it aids researchers in model debugging, data collection, trust building and human decision-making. There are two main approaches of XAI methods; intrinsic XAI, and post-hoc XAI. Intrinsic XAI approaches provide explanations to the structure and functioning of the model, which limit its application to a specific type of AI techniques. Post-hoc approaches provide explanations to the model's final decision by analysing the set of input and the set of output; thus, it is applicable to any type of models. XAI methods are also divided into global or local. Global XAI methods aim to explain the general characteristics of the models by analyzing all of its decisions. Whilst local XAI methods focus on giving explanations to individual decisions that the model has made.

The large scale development of the Internet of Things (IoT) has caused significant new challenges in the areas of cyber defence. IoT deployments are Large number of Internet-connected devices generates huge amount of traffic and digital data generated, which challenges existing solutions' ability to detect and prevent cyberattacks [164]. Therefore, AI-based IDS is a prominent method to deal with such large-scale data [129]. A huge body of researches have been focused on finding the best IDS in IoT enviroments [164]. Additionally, XAI methods would be applied in this field to enhance the performance, interpretability and explanability of AI-based intrusion detection models in the heterogeneous IoT networks.

XAI methods have been created and applied to many applications and fields. The most popular works focus on model explanations for computer vision [165], Natural Language Processing (NLP) [166] and voice analysis [167]. Of particular note, the authors of [166] developed and deploy a framework, LIME, in the field of NLP and [168] used SHAP to give explanations in the field of biology. In contrast, the utilization of XAI methods in the field of cyber defence is very new and rare.

In the field of cyber defence, AI is mainly utilised to build a model for automated detection. Numerous researchers have shown the prominence of AI-based IDS. Of note amongst these, Rahul et al. [129] used a benchmark dataset to compare the performance of several classical ML algorithms and Deep Neural Network (DNN) models. In this experiment, AI-based IDSs obtained excellent performance, particularly those built upon DNNs outperformed other classical ML algorithms. Andalib and Vakili [136] proposed an IDS combining three different types of ML techniques. Based on their observations, ML models have a short training time and only need low computational resources [136]. Despite the prominent performance, the AI-based intrusion detection models are becoming more complex and their functions and outcomes are difficult to interpret. This makes them into black-box models and hinders their applications in critical sectors including cyber defence.

In order to confront this problem, several works related to explainable AI-based IDS have emerged. Islam et al. [150] proposed a method to gather and utilize domain knowledge to automate the defence response and improve the explainability of the IDS model. In the experiment, domain knowledge (including fundamental principle of Confidentiality, Integrity, and Availability (CIA)) is instilled into the ML models. This was analyzed with the CIDS2017 dataset. Results from this experiment highlighted an increase in generalizability and explainability of the models, which promoted the utilization of AI-based IDS in IoT networks. Marino et al. [151] produced explanations for incorrect samples using an adversarial approach. This approach works by finding the minimum modification to correctly label the incorrect records then visualize the most critical features to explain the model's wrong decision. Le et al. [152] propose a IDS built upon an RNN and explain the outcomes of the model to generate Software Defined Networking (SDN) flow rules. They use a linear regression model as a local surrogate describes in LEMNA [153], the k most relevant features are chosen to generate network access control policies.

3.3 Proposed Methodology

This section introduces the underlying mathematical models of the IDS and SPIP framework. Although our main focus in this research is the interpretation of black-box model, the performance of IDS is also important. Valid explanations can only be extracted from well-performing models with good predictions. If the model perform poorly, the explanations generated by the framework may be obscure and bias. Therefore, Deep Neural Networks (DNNs) are chosen as the underlying algorithm to detect attacks due to its prominent performance that is shown in Section 3.2. The proposed framework applies various XAI methods to produces reasonable and intuitive explanations that would be used interpret predictions of the model and study the quality of datasets as well as the characteristics of many attack classes.

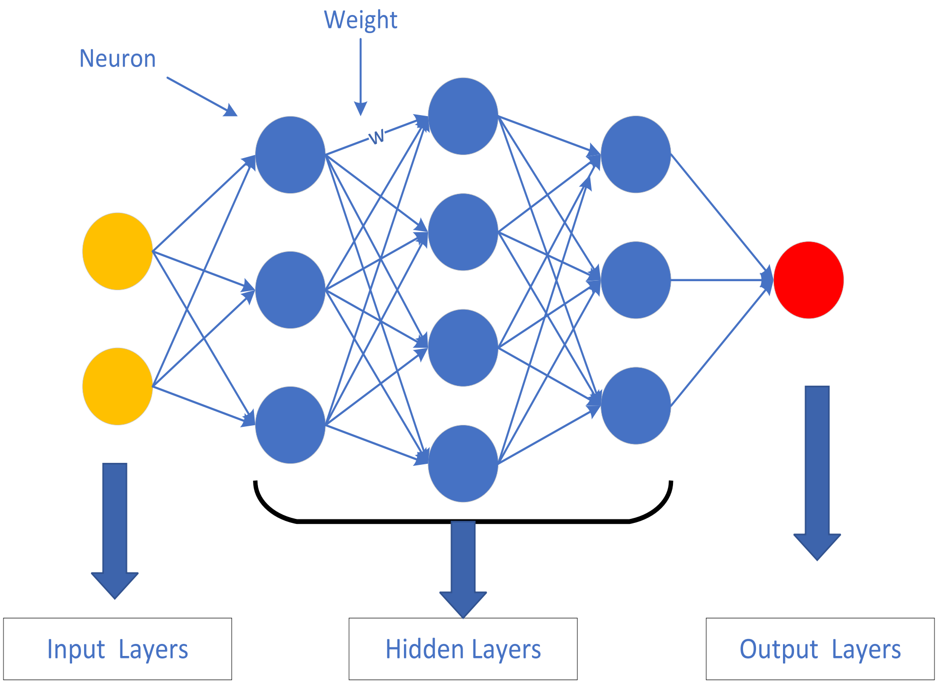


Figure 3.1: Deep Neural Network Structure

3.3.1 Deep Neural Network

Deep neural network (DNN) is nonlinear model which is designed to have a complex structure in order to work with complex computation and problems. The networks consist of layers and each layer comprises of many neurons. There are three different layers, namely input layer, hidden layer and output layer. Input data are first fed into neurons in input layer, the model then forward information into neurons of hidden layers through a link, then neurons in output layer will be used to predict a class for the specified input data. An example of DNN consisting of the three types of layers are shown in 3.1. As the information flows from left to right, the process mentioned above is called feed-forward.

To further discuss feed-forward process, each link that connect neurons between different layers has a weight which is used to get value of sum function of each neuron. The value of sum function of each neuron is compared to the *bias* to determine whether

Algorithm 1: Feed-forward Process

```
Let K$_i$ be the number of nodes in layer i;
w$^{n,m}_{i,j}$ be the link between node i and j;
noL be the number of layers;
$\zeta$ and $\alpha$ is the sum function and activation function of each neuron, respectively.
for each layer m = 2, 3, ...,noL do
    for each neuron j = 1, 2, ..., K$_i$ do
        $\zeta \leftarrow \sum_i w^{m,m-1}_{i,j}$
        Calculate activation function: $\alpha(\zeta)$
```

the neuron will be activated or not. Then activation function of each neuron are computed, which uses value of sum function as a parameter. Many activation functions are invented and utilised for different problems, such as *"sigmoid", "rectified linear unit", "softmax", binary step* functions, each with its own advantages. This process happens as information flows from input layer to output layer. Once the activation function of neurons in output layer are computed, the loss function is used to determine the difference between predicted outcome and the actual outcome of input data [65]. The Binary Cross-Entropy is calculated as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p(y_i)) + (1-y_i)log(1-p(y_i))) \tag{3.1}$$

where

- N: number of inputs
- y: the label (1 for attack instances and 0 for normal instances)
- p(y): the predicted probability of the instance being attack for all N points

Once the loss function is known, the reverse process would be started to reduce the loss function by updating the weights. This is called backpropagation, as weights are updated from the output layer to the input layer, from right to left. The weights

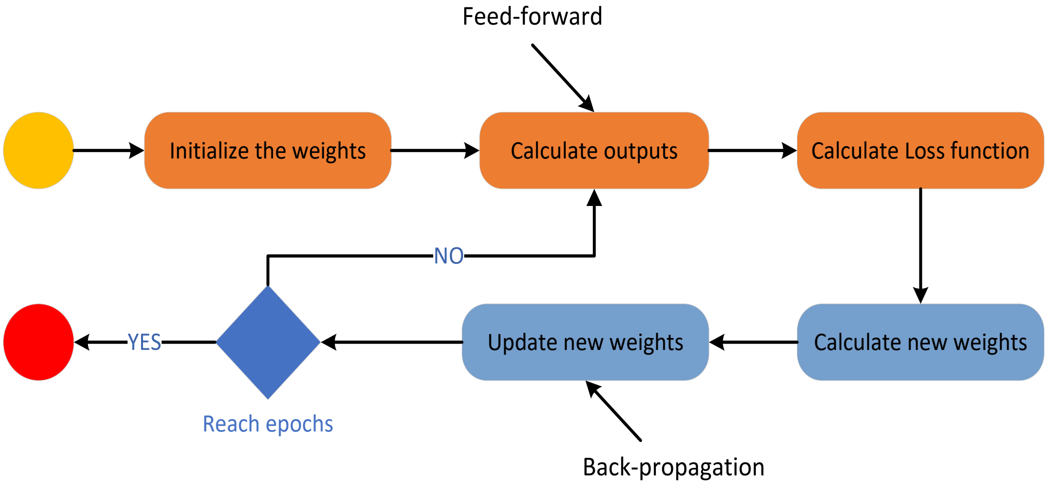


Figure 3.2: Deep Neural Network Training Process

between neurons are updated according to how much they are responsible for the error in the loss function, and the learning rate is set to decide how much we update the weights. The learning rate needs to be optimal. While a high learning rate is fast but may lose the global minimum, a low learning rate is slower but more likely to reach the global minimum.

Using gradient descent, the loss function can be minimised, and a perfect model will have a lost function of 0 [65]. There are a number of algorithms for optimisation techniques for gradient descent. To find the best algorithm, *RMSProp* and *Adam* will be tested and evaluated:

- RMSProp, also know as Root Mean Square Propagation, is an adaptive learning algorithm. It takes the exponential moving average to minimize the aggressive, monotonically decreasing learning rate [169]:

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t + \varepsilon)^{\frac{1}{2}}} \frac{\delta \mathcal{L}}{\delta w_t} \tag{3.2}$$

$$v_t = \beta_{v_{t-1}} + (1-\beta)[\frac{\delta \mathcal{L}}{\delta w_t}]^2 \tag{3.3}$$

where:

  - w_t: weights at time step t
  - α_t: learning rate at time step t
  - $\delta\mathcal{L}$: derivative of loss function
  - v_t: sum of square of past gradients
  - β: exponential decay rates (β=0.9)

- Adam is an advanced stochastic optimization method that was first introduced by Kingma et al. in [170]. It is widely-used in the field of deep learning [171] as it is computationally efficient and has little memory requirements.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\frac{\delta\mathcal{L}}{\delta w_t} \tag{3.4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)[\frac{\delta\mathcal{L}}{\delta w_t}]^2 \tag{3.5}$$

  where:

  - m_t: aggregate of gradients at time t
  - β_1, β_2: exponential decay rates (β_1 = 0.9, β_2 = 0.999) [170]

After updating new weights for all connections between neurons, the feed-forward process will be restarted to calculate the new loss function. The training process would repeat a number of times which is called *epochs*. An epoch is a period taken to pass the entire dataset through the network once [172]. We need to optimise the number of training epochs to avoid underfitting and overfitting. While underfitting happens when the model is insufficiently trained, overfitting occurs when the model performs well on existing data but cannot correctly classify unknown data [173].

Moreover, we also apply the dropout rate to avoid overfitting. In other words, dropout would prevent neurons from extracting the same hidden features from input data from being activated. If the duplicated features only occur in the training data, this would lead to overfitting; then a dropout rate is needed to prevent this. In general, the training process of DNN is demonstrated in Figure 3.2.

3.3.2 Explainable Artificial Intelligence (XAI) methods

Model-agnostic XAI techniques can be classified into two categories, including local and global explanation methods. Local interpretability would use only one data record to generate the explanation for a single model's decision. This type of explanation is useful for users and people that are affected by the model's decision in helping them to trust the model and verify a single decision that affects them. Global interpretability would take the whole dataset as input to generate explanations, demonstrating the black-box model's overall structure and functioning. This technique is beneficial for researchers and developers to analyse and detect any bias to improve the model's performance. Moreover, in the case of IDS deployments, global XAI methods are essential in studying and analysing different types of cyberattacks based on features of network traffics.

In this chapter, different XAI methods are used to study the functioning and predictions of the proposed IDS both locally and globally. On the one hand, we apply Shapley Additive Explanations (SHAP) [98] and Individual Conditional Expectation (ICE) [86] techniques for visualising individual prediction made by the model. On the other hand, Permutation Feature Importance (PFI) [97], SHAP [98]. Partial Dependence Plot (PDP) [84] are utilised as global XAI methods to find the customised set of relevant input features, which can be used to explain the functioning of the proposed model as a whole. Those XAI techniques are discussed and detailed as follows.

3.3.2.1 Shapley Additive Explanations (SHAP)

SHAP was proposed by Lundberg et al. [98] based on the game theoretically optimal Shapley values in [110]. Therefore, SHAP has a strong theoretical foundation. SHAP is a combination of Local Interpretable Model-agnostic Explanations (LIME) [75] and Shapley value [110]. Thus, we will briefly introduce LIME and Shapley's value before analysing SHAP.

LIME derives from the concept of local surrogate models. In the concept of surrogate model, an interpretable model is chosen to be trained and approximate the predictions of the black-box model that need to be explained. Different from global surrogate models, LIME's key concept is fitting an interpretable model around a specific instance to visualise significant features of that data instance in contribution towards an attack or normal prediction of the model. In LIME, the surrogate model can be of any type of interpretable model as long as it is a good approximation of the black-box model's predictions locally [75]. In LIME, the explanation of an observation x is computed by the following formula:

$$E(x) = \underset{g \in G}{\operatorname{argmin}} \left\{ L\left(f, g, d^{x}\right) + \Omega(g) \right\} \tag{3.6}$$

In equation (3.6), g denotes an interpretable model and G represents the class of all possible explanation models. These can be any type of interpretable ML whose results can be understood by a human. The original model is denoted by f and the distance between the sample and original data is presented as d^{x}. Thus, L(f, g, d^{x}) is the loss function that measures the difference in decisions of the surrogate model and the original model. LIME aims to calculate this loss function L. Additionally, Ω*(g)* denotes the complexity of model g and it is defined by human. For example, Ω*(g)* can be point for the depth of the decision tree model [75].

The LIME algorithm begins by creating non-existing data samples around the chosen instance of interest *x*. Then a local surrogate model g is trained on this data samples as the loss function L and the complexity Ω are minimized. Eventually, a local model denoted by *E(x)* can explain the given decision of instance *x*. The Shapley value is a technique in the cooperative game theory method. The value was named after its inventor Lloyd Shapley as he introduced it in [110]. Shapley value has a strong theoretical foundation to measure the importance of each player in a collaborative game and indicate how much each player has contributed to the success. Specifically, each feature's value has its own Shapley value that represents its average contribution to the model's outcome. This value is computed and summed over all combinations of feature values for a given set of features and calculated predictions:

$$\phi_j(val) = \sum_{S \subseteq M \backslash \{j\}} \frac{|S|!\,(m - |S| - 1)!}{m!} [val(S + j) - val(S)], \quad j = 1 \ldots m \tag{3.7}$$

where:

- S is the subset of features used and |S| is its number.
- M is the complete set of features and m is its size.
- j is the *j*th feature the vector of the feature values of the instance to be explained
- *val(S)* and *val(S+j)* are the functions assigning only subset of features S and subset of features S with *j*th feature present, respectively.

The Shapley value has four meaningful properties; **Efficiency, Symmetry, Dummy** and **Additivity**. The property of **Efficiency** states that the feature contributions must add up to the prediction for a single instance which equals to the difference between the actual output and the average prediction:

$$\sum_{i=1}^{m} \phi_i(val) = val(M) \tag{3.8}$$

Symmetry: Considering two different features *i*th and *k*th in the subset of features S, so that:

$$val(S + i) = val(S + k), \quad S \subseteq M \backslash \{i, k\} \tag{3.9}$$

Equation 3.11 indicates that two features *i*th and *k*th contribute equally to all possible coalitions. Therefore, the contribution of two features should be equal and the function ϕ is symmetric:

$$\phi_i(val) = \phi_k(val) \tag{3.10}$$

Dummy: if contribution of a given feature does not change the prediction, its Shapley value should be equal to zero:

$$val(S + i) = val(S), \quad S \subseteq M \backslash \{i\} \tag{3.11}$$

then:

$$\phi_i(val) = 0 \tag{3.12}$$

Additivity: the gain from combining two functions *val* and *val′* is equal to sum of individual gains from each function for every feature ith:

$$\phi_i(val + val') = \phi_i(val) + \phi_i(val') \tag{3.13}$$

SHAP was first introduced in 2017 [98] as a unified approach for explaining black-box models' predictions. SHAP is a type of relevance-based approach that measures the importance of each feature by computing its contributions to the model's decision. SHAP can connect LIME and Shapley value by representing Shapley value explanation as a linear model. SHAP value can be calculated using individual data instances. For example, explanation for an instance x can be calculated using the following formula:

$$g\left(z'\right) = \phi_0 + \sum_{i=1}^{m} \phi_i z_i', \quad z' \in \{0,1\}^m \tag{3.14}$$

where:

- m is the size of all possible coalitions.
- g is the explanation model.
- z' is simplified features. The 1 in z$'$ indicates that features in the new data are the same as those of the original data - the instance z.
- ϕ_i is the Shapley value for feature $i^{th}$of instance x, hence indicates the impact of feature i on the model's prediction.

A key advantage of SHAP over LIME is that effects are distributed fairly in SHAP yet behaviour of the model is assumed to be linear locally in LIME. Specifically, LIME weighs the instances based on their distance to the original instance. Meanwhile, instances in SHAP are weighted so that the coalition is in the Shapley value estimation. Shapley compliant weighting is calculated by a method called SHAP kernel [98]:

$$\pi_x\left(z'\right) = \frac{(m-1)}{\begin{pmatrix} m \\ |z'| \end{pmatrix} |z'| \left(m - |z'|\right)} \tag{3.15}$$

where:

- m is the size of all possible coalitions.
- $|z'|$ is the total number of present features in instance z'

To conclude, the process of calculating SHAP values for an instance x is divided into five steps:

1. Sample data $z'_k \in \{0,1\}^m$, k $\in$ {1 ... K}
2. Get prediction for each z'_k. This requires 2 steps, including converting each z'_k to the original feature space, and then applying model *f(*h_x*(*z'_k*)). f(x)* represents original model, and $h_x(z') = z$ where h$_x$: {0,1}$^m \rightarrow \mathbb{R}$ is used to get valid data instances from coalitions of feature values.
3. Calculate the weight for each z'_k using equation (3.15)
4. Train the linear model g by optimizing the loss function L and using Z as the training data:

$$L(f, g, \pi_x) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_x(z') \tag{3.16}$$

5. Return ϕ_k as Shapley values and coefficients from the linear model calculated in equation (3.16).

3.3.2.2 Permutation Feature Importance (PFI)

PFI is another type of relevance-based model-agnostic method that computes the changes in prediction error of a fitted model when permuting the input feature values. This technique shows the importance of each feature by breaking the relationship between the feature and the desired output. Therefore, the model's error would increase if the feature of interest is considered important by the model. PFI was first introduced in 2001 by Breiman [96]. Based on this concept, Fisher et al. created a

Algorithm 2: Permutation Feature Importance algorithm

```
$f$: trained model, $X$: feature matrix, $y$: target vector, $\mathcal{E}(y, f)$ error measure
Compute the original model's error: $e^{orig} = \mathcal{E}(y, f(X))$
for  each $i = \{1...k\}$ do
    Create $X^{perm}$ by permuting feature $i$ in the data $X$ to break the relationship
      between feature $i$ and true outcome $y$.
    New error value $e^{perm} = \mathcal{E}(y, f(X^{perm}))$
    Calculate the PFI value for feature $i^{th}$: $S_i = \frac{e^{perm}}{e^{orig}}$
Sort and present features in descending order of value $S_i$
```

model-agnostic version of the PFI named *Model Class Reliance* in 2018. The steps to calculate Permutation Feature Importance are shown in 2 [97].

3.3.2.3 Partial Dependence Plot (PDP)

Partial Dependence Plot (PDP) is one of the most widely adopted visualisation-based XAI methods. It was first introduced by Friedman (2001) to show the relationship between feature values and the black-box model outcome by visualising on the plot [83]. PDP is a global model-agnostic approach, which means that it considers all records and demonstrates the global relationship between a specific feature and the prediction.

$$\mathcal{PD}(x_s) = E(x_c)[f(x_s, x_c)] = \int f(x_s, x_c) dP(x_c) \tag{3.17}$$

where:

- f refers to the ML model
- $x_s \in S$ - set of input features for which PDP function is plotted.
- $x_c \in C$ - set other input features *f*, which means that: $x_c = x \setminus x_s$
- *$dP(x_c)$* is the marginal distribution of x_c

According to equation (3.17), PDP function marginalise model f outcome over the distribution of set C in order to shows the relationship between the features in set S and model f output. Another way to compute the partial function $\mathcal{PD}(x_s)$ by using Monte Carlo method [74]:

$$\mathcal{PD}(x_s) = \frac{1}{k}\sum_{i=1}^{k} f(x_s, x_c^i) \tag{3.18}$$

where:

- k refers to the total number of data records
- x_c^i are feature values from the set of features C

Equation (3.18) demonstrates the average marginal effect of given feature values in a set of features S on the model outcome. Additionally, PDP assumes that there is no correlation between features in S and features in C. However, this assumption is likely violated in real life; thus, the function $\mathcal{PD}$ would take invalid data points as arguments and demonstrate misleading explanations. Moreover, PDP can have a hidden heterogeneous effect as it only visualise the average marginal effect. Specifically, each data point can have a positive or negative association with the model's prediction; thus, only presenting the average effects of all points would hide such relationships [74].

3.3.2.4 Individual Conditional Expectation (ICE)

Individual Conditional Expectation (ICE) is calculated similarly to PDP, but it is designed to inspect individual observations of the model. Compared to one line overall in PDP, ICE curves display one line for each record that demonstrates the record's prediction changes when a feature changes. In other words, PDP represents the average of all lines in an ICE plot. PDP covers the heterogeneous relationships created by interactions between features; therefore, it only works well if these interactions are weak. ICE uncovers such relationships, hence providing more insight than PDP. ICE would take each record in the observations set $\{(x_s^{(i)}, x_c^{(i)})\}_{i=1}^{k}$ and plot the curve $f_s^{(i)}$ against

$x_s^{(i)}$. Thus, the relationship of the feature with prediction is shown for k observations, and the average of all curves results in PDP.

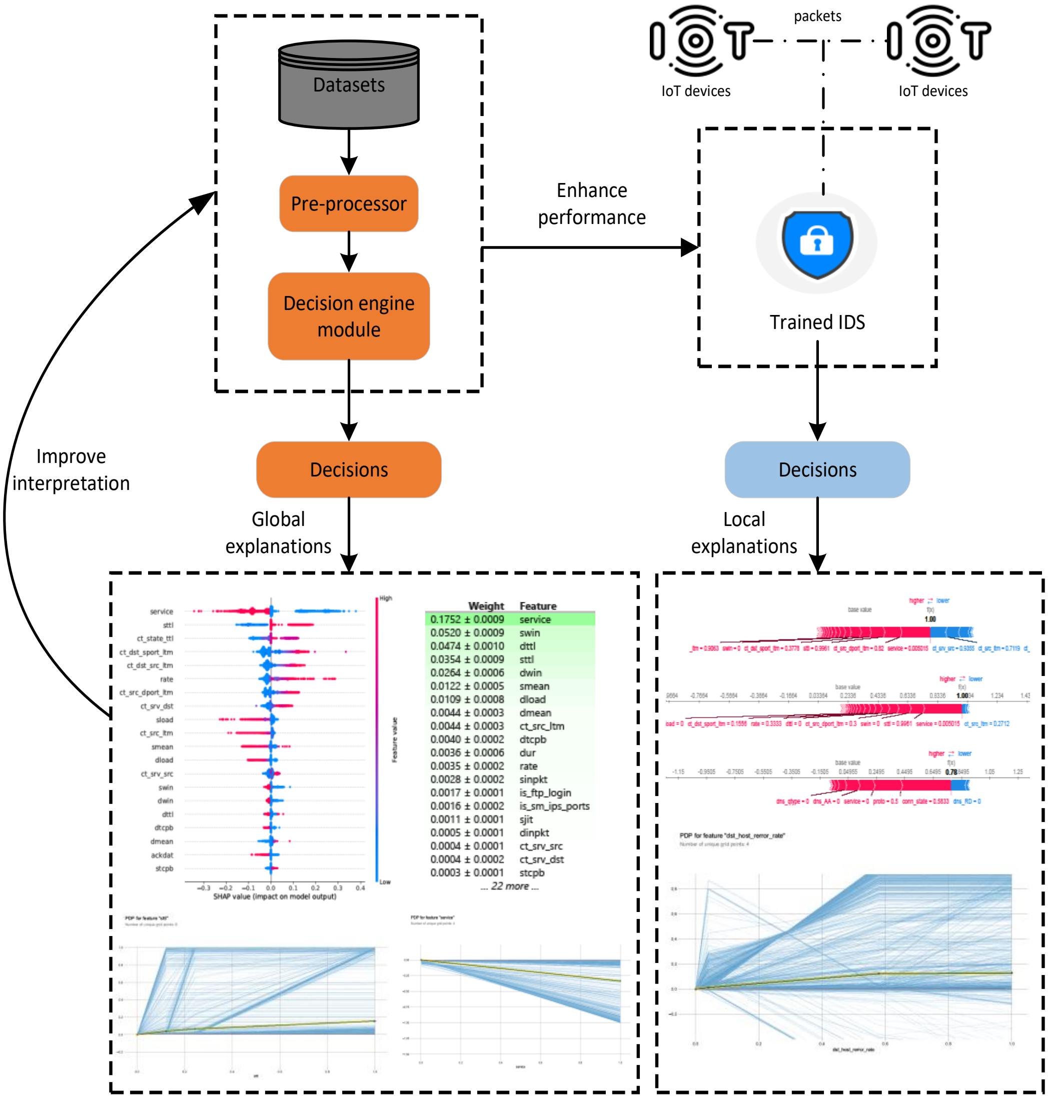
3.4 Proposed Method

This section demonstrates the proposed framework for an explainable AI-based IDS platform. In IDS, identifying malicious behaviour is only the first step because understanding such a decision is crucial for a solution. An insight into the decision helps administrators identify the part of the network, features, and security policies compromised by attackers [77]. With the information provided by XAI, the IDS operator can give the correct actions, whether it is to debug the IDS model or apply new security policies to prevent the same attacks in the future.

The proposed framework aims to develop an effective explainable AI-based IDS, therefore, various XAI methods are used to generate local and global explanations. The local approach focuses on explaining single network traffic that was classified as intrusion; thus, it increases the security expert's and user's trust in the IDS. Meanwhile, the global approach gives insights into the complex functioning of the IDS, hence helping the IDS operators to improve the model's efficiency and providing the experts with deep knowledge about cyber attacks.

3.4.1 Local Explanation

The proposed framework utilises two different methods to generate local explanations, including Shapley Additive Explanations (SHAP) and Individual Conditional Expectation (ICE). Shapley values of individual prediction measure the effect of each input feature on the model's prediction. The explanations are visualised to be more intuitive. Additionally, XAI recipients can easily find how that value of a particular feature affects the predicted probabilities made by the IDS. Meanwhile, ICE inspects each data record and displays one line for each instance, demonstrating the prediction's changes



Weight	Feature
0.1752 ± 0.0009	service
0.0520 ± 0.0009	swin
0.0474 ± 0.0010	dttl
0.0354 ± 0.0009	sttl
0.0264 ± 0.0006	dwin
0.0122 ± 0.0005	smean
0.0109 ± 0.0008	dload
0.0044 ± 0.0003	dmean
0.0044 ± 0.0003	ct_src_ltm
0.0040 ± 0.0002	dtcpb
0.0036 ± 0.0006	dur
0.0035 ± 0.0002	rate
0.0028 ± 0.0002	sinpkt
0.0017 ± 0.0001	is_ftp_login
0.0016 ± 0.0002	is_sm_ips_ports
0.0011 ± 0.0001	sjit
0.0005 ± 0.0001	dinpkt
0.0004 ± 0.0001	ct_srv_src
0.0004 ± 0.0002	ct_srv_dst
0.0003 ± 0.0001	stcpb

... 22 more ...

Figure 3.3: Overview of the structure of SPIP framework

according to a specific feature's changes. Therefore, ICE can support Shapley's values very well in showing the impact of a feature on the final prediction.

3.4.2 Global Explanation

The proposed framework utilises three different methods to generate global explanations, including Partial Dependence Plot (PDP), Permutation Feature Importance (PFI) and SHAP. PDP visualises the global impact of specific features on the predictions of a fitted ML model. This helps security experts to study the importance of each feature in a particular benchmark dataset. PFI computes the change in prediction error of a fitted model over all possible permutations of the feature's values; thus, it shows the global impact of each feature on the model's predictions.

PFI typically shows the twenty most important features from the set of input features; therefore, it would be useful in feature selection. SHAP can also be used to show the most important features by combining Shapley values. Calculating SHAP for each data instance would generate a matrix whose rows represent for instances and columns denote features. Then, the importance score of each feature can be obtained by computing the average of the absolute Shapley values per feature across this matrix:

$$\mathcal{A}_j = \sum_{i=1}^{k} ||\phi_j(x_i)|| \tag{3.19}$$

where:

- $\mathcal{A}_j$ is the average Shapley value of the j^{th} feature
- k refers to the number of instances in the dataset
- $\phi_j(x_i)$ is the Shapley value of the j^{th} feature in the i^{th} instance

After obtaining these values, the features are typically sorted based on the importance score and the twenty most important features are shown.

3.5 Experimentation and Results

In this section, the experiment configurations, including the datasets used and the structures of the models, are discussed. We also evaluate the performance of different classifiers and the proposed explainable AI-based IDS framework. As discussed above, our framework generates both global and local explanations using multiple XAI methods. The experiments aim to evaluate the efficiency of generated explanations. Additionally, different classifiers are customised to utilise the results obtained from the proposed framework, improving the model's performance and giving more insight into the datasets used and the functioning of the AI-based IDS.

3.5.1 Datasets

Several statistical analyses have revealed the drawbacks of the benchmark dataset KDD-CUP99 and shown that the dataset may adversely affect the performance, evaluation and comparison of different classifiers. NSL-KDD is a purified version of KDD-CUP99 as it has significant improvement. Firstly, redundant and duplicate records are removed to prevent the IDS from being biased [174]. Secondly, the number of selected records from each difficulty-level group is inversely proportional to the percentage of records in the KDD-CUP99. As a result, the evaluation of IDS is more accurate because the range of classification rates is wider [174]. Lastly, the NSL-KDD dataset contains a sufficient and reasonable number of records to ensure reliable results from different classifiers. Each data instance has a total of 41 features and one label classifying it into attack or normal network flow. The features can be basic, content-related, time-related or host-based traffic features [175]. There are three datasets in NSL-KDD, including (1)KDDTrain+ for training, (2)KDDTest+ for testing, and (3)KDDTest-21 for advanced testing. Researchers designed 21 machine learning models to evaluate the NSL-KDD datasets. Then, they removed all records in KDDTest+ that were correctly labelled by 21 models and made a new dataset from the leftover called KDDTest-21 [174].

Table 3.1: List of symbolic and binary features in the three datasets

Dataset	Symbolic features	Binary features
NSL-KDD	protocol_type, service, flag	land, logged_in, root_shell, su_attempted, is_host_login, is_guest_login, logged_in
UNSW-NB15	proto, state, service	is_sm_ips_ports, is_ftp_login
ToN_IoT	conn_state, dns_query, ssl_version, ssl_cipher, ssl_subject, ssl_issuer, http_method, http_uri, http_version, http_orig_mime_types, http_resp_mime_types, weird_name, weird_addl, type	dns_AA, dns_RD, dns_RA, dns_rejected, ssl_resumed, ssl_established, weird_notice

Moustafa and Slay (2015) created the UNSW-NB15 dataset in a synthetic setting built in the UNSW cyber security lab. UNSWNB-15 is a modern NIDS benchmark dataset as the creators utilize novel methods to generate modern normal and synthetic attack network activities [27]. Unlike NSL-KDD/KDDCup-99 datasets, UNSW-NB15 contains new attack and standard network traffic patterns. IXIA is utilized to generate network traffic then Argus, Bro-IDS tools and twelve algorithms are used to extract a total of 47 features from the generated activities. Depending on the IXIA tool's report, each record has 2 labels including (1) *attack_cat*: name of each attack category and (2) *label*: 0 for normal, 1 for attack records. UNSW-NB15 has multiple attacks and up-to-date information of packets, which are significant advantages over old datasets.

Moustafa [3] developed a new dataset to validate Industrial IoT (IIoT) systems named the ToN_IoT dataset. The dataset contains telemetry data, which are collected from IoT devices, to detect intrusions that manipulate IoT devices [176]. The IoT telemetry data was generated in a testbed environment with three layers *Edge, Fog* and *Cloud* to represent real-life data from current production IoT/IIoT networks. The NSX-VMware platform was utilized to provide the features of Software-defined Network (SDN) and Network Function Virtualization (NFV) to manage the interaction between the three layers [176]. ToN_IoT was generated on a realistic simulation of an IoT environment, incorporating both normal and various types of attacks in IIoT applications.

Table 3.2: List of attack types in NSL-KDD

Class of attack	Attack types
probe	ipsweep, nmap, portsweep, satan, saint, mscan
DoS (Denial of Service)	back, land, neptune, pod, smurf, teardrop, apache2, udpstorm, processtable, mailbomb
u2r (user to root)	buffer_overflow, loadmodule, perl, rootkit, xterm, ps, sqlattack
r2l (root to local)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, snmpgetattack, named, xlock, xsnoop, sendmail, httptunnel, worm, snmpguess

3.5.2 Data Pre-processing

Each dataset above has several symbolic and binary input features. Names of such features are listed in table 3.1. Symbolic features from these datasets are converted using a label encoder and binary features remained unchanged. Other input features in the three datasets are continuous which can be integer or float. In this experiment, we use min-max normalization method so that all continuous features are in the range [0,1] after being scaled:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.20}$$

There are 17 classes of attack in the three datasets. Among those, *Denial of Service (DoS)* attack appears in all three datasets. *Probe* attack in NSL-KDD is equivalent to *Analysis* in UNSW-NB15 and *Scanning* in ToN_IoT. Additionally, both UNSW-NB15 and ToN_IoT have the *Backdoor* attack class. In the NSL-KDD dataset, classes of attack are categorised into different types that are shown in table 3.2. The *label* column in NSL-KDD contains these attack types' names, therefore, we must convert them into the original attack classes' names before using the NSL-KDD dataset.

3.5.3 Evaluation Metrics

Accuracy, precision, recall, and F1-score are evaluation indicators to evaluate the model's performance. These statistical measures are calculated from ground truth values including True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). These evaluation indicators explain the performance of the model; thus, they provide feedback to make improvements and design an IDS in IoT networks with desirable performance. The definition of ground truth values are listed as follow:

- **True Positive (TP)**: number of instances correctly classified as the target outcome
- **True Negative (TN)**: number of instances correctly classified as others
- **False Positive (FP)**: number of instances wrongly classified as the target outcome
- **False Negative (FN)**: number of instances wrongly classified as others

Based on above values, evaluation indicators are calculated as follow:

1. **Accuracy**: ratio of the number of correctly classified instances to the total number of instances in the test set.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

2. **Precision**: ratio of the number of instances correctly classified as the target outcome to the number of instances classified as the target outcome

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{20}$$

3. **Recall**: ratio of the number of instances correctly classified as the target outcome to the total number of target records.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{21}$$

4. **F1-score**: this value measures precision and recall at the same time by using the harmonic mean in place of the arithmetic mean

$$F1 \text{ -score } = \frac{2 * \text{ Recall } * \text{ Precision}}{\text{Recall } + \text{ Precision}} \tag{22}$$

3.5.4 Computer Environment

The models were developed using the Python programming language on Microsoft Windows 10 Home Version 20H2 . These were evaluated on a system with with 16 GB RAM. In this experiment, we use TensorFlow - a major deep learning framework to train and evaluate the model. TensorFlow is GPU-accelerated and this framework utilised the NVIDIA GeForce GTX 1060 3GB on the evaluation system to effectively train the DL-based models. All code in this study can be found at [177].

3.5.5 Intrusion Detection System

The first stage of this work is to build and train different types of classifiers. Firstly, binary classifiers are built to detect attacks without identifying the class of such attacks. From there, in this work we apply global XAI methods (SHAP, PDP) to study the most important features in each dataset and select the most relevant features to build a more efficient model. Secondly, we build one-vs-all classifiers to detect each class of attack separately. After obtaining results from fitted one-vs-all classifiers, the proposed framework will generate both local and global explanations, giving insights into each feature's importance and the functioning of models. Lastly, we build target-vs-normal classifiers that distinguish between a targeted attack class and the normal traffic. This will help domain experts in analysing the distinctive characteristics of each attack class or attack type.

Although the focus of this research is interpretation of the black-box model, we need to build an effective IDS in order to generate meaningful explanations. Therefore, we test and evaluate numerous parameters to find the optimal values for deep learning architecture, including optimizer (RMSProp and Adam), hidden layers (2-5 layers),

Table 3.3: Results about Binary Classifiers

Dataset	Accuracy	Precision	Recall	F1-score
NSL-KDD	0.811	0.921	0.732	0.815
UNSW-NB15	0.866	0.811	0.988	0.891
ToN_IoT	0.873	0.784	0.880	0.829

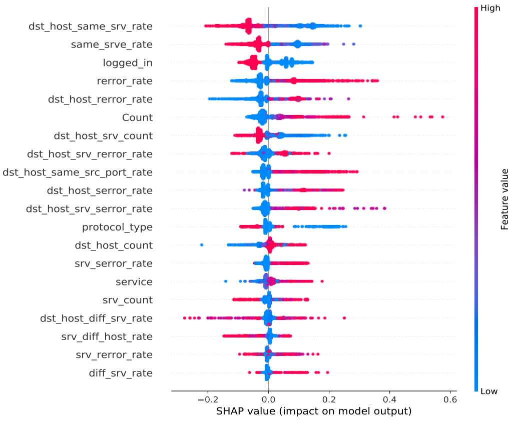
number of neurons in each hidden layer (20, 30, 50), training epoch (10, 20, 50), dropout rate (0.1, 0.2, 0.3), learning rate [0.001, 0.002, 0.003].

Eventually, we obtain an optimised set of parameters as follows. The models based on fully connected networks with ReLU activation. The input dimension of each classifier depends on the dataset used as it equals the number of input features. Each classifier has 3 hidden layers, and each hidden layer contains 50 neurons. The output dimension of each classifier is 2. The TensorFlow framework is used to effectively build the intrusion detection models. We use Adam optimizer, learning rate of 0.001 and dropout rate of 0.1. The batch size is 32 with a total of 10 epochs.

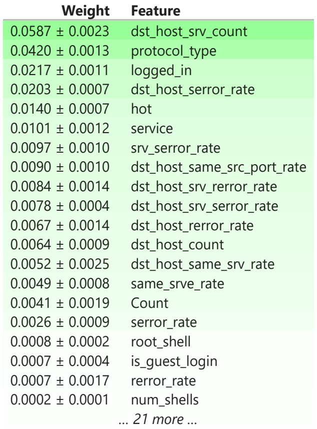
3.5.6 Binary Classifiers

We construct and train binary classifiers on the three datasets using the complete set of input features. Table 3.3 shows the evaluation indicators obtained from these classifiers. These values indicated that the performance of the models is sufficient to apply the proposed XAI framework and study the relevant features that models have learned to identify malicious activities. From the fitted model, the framework calculates the importance score of each feature using SHAP and PFI. The most important features of attacks that the model has learned from the three datasets are shown in Figure 3.4.

As shown in Figure 3.4, lists of the top 20 important features that are generated by SHAP and PFI are quite different in all datasets. The outputs from PFI are easier to understand since each feature has a simple numeric value that indicates its overall impact on the model's decision. Therefore, the comparisons between features are relatively simple, which makes PFI's lists suitable for lay-users to understand. However, explanations generated by PFI are so simple that they only calculate the medium per-

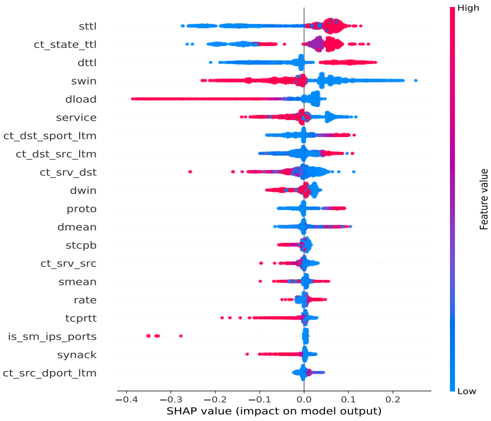


(a) NSL-KDD features extracted by SHAP

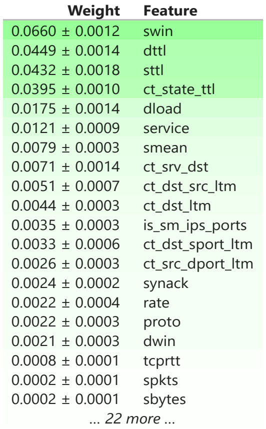


Weight	Feature
0.0587 ± 0.0023	dst_host_srv_count
0.0420 ± 0.0013	protocol_type
0.0217 ± 0.0011	logged_in
0.0203 ± 0.0007	dst_host_serror_rate
0.0140 ± 0.0007	hot
0.0101 ± 0.0012	service
0.0097 ± 0.0010	srv_serror_rate
0.0090 ± 0.0010	dst_host_same_src_port_rate
0.0084 ± 0.0014	dst_host_srv_rerror_rate
0.0078 ± 0.0004	dst_host_srv_serror_rate
0.0067 ± 0.0014	dst_host_rerror_rate
0.0064 ± 0.0009	dst_host_count
0.0052 ± 0.0025	dst_host_same_srv_rate
0.0049 ± 0.0008	same_srve_rate
0.0041 ± 0.0019	Count
0.0026 ± 0.0009	serror_rate
0.0008 ± 0.0002	root_shell
0.0007 ± 0.0004	is_guest_login
0.0007 ± 0.0017	rerror_rate
0.0002 ± 0.0001	num_shells
... 21 more ...	

(b) NSL-KDD features extracted by PFI

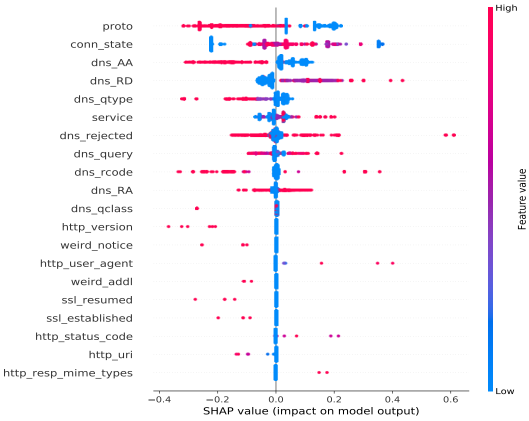


(c) UNSW features extracted by SHAP

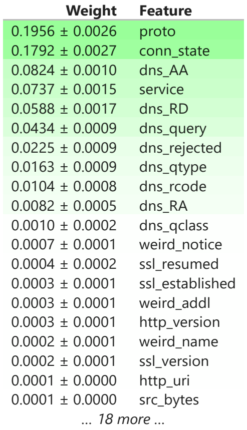


Weight	Feature
0.0660 ± 0.0012	swin
0.0449 ± 0.0014	dttl
0.0432 ± 0.0018	sttl
0.0395 ± 0.0010	ct_state_ttl
0.0175 ± 0.0014	dload
0.0121 ± 0.0009	service
0.0079 ± 0.0003	smean
0.0071 ± 0.0014	ct_srv_dst
0.0051 ± 0.0007	ct_dst_src_ltm
0.0044 ± 0.0003	ct_dst_ltm
0.0035 ± 0.0003	is_sm_ips_ports
0.0033 ± 0.0006	ct_dst_sport_ltm
0.0026 ± 0.0003	ct_src_dport_ltm
0.0024 ± 0.0002	synack
0.0022 ± 0.0004	rate
0.0022 ± 0.0003	proto
0.0021 ± 0.0003	dwin
0.0008 ± 0.0001	tcprtt
0.0002 ± 0.0001	spkts
0.0002 ± 0.0001	sbytes
... 22 more ...	

(d) UNSW features extracted by PFI



(e) ToN_IoT features extracted by SHAP



Weight	Feature
0.1956 ± 0.0026	proto
0.1792 ± 0.0027	conn_state
0.0824 ± 0.0010	dns_AA
0.0737 ± 0.0015	service
0.0588 ± 0.0017	dns_RD
0.0434 ± 0.0009	dns_query
0.0225 ± 0.0009	dns_rejected
0.0163 ± 0.0009	dns_qtype
0.0104 ± 0.0008	dns_rcode
0.0082 ± 0.0005	dns_RA
0.0010 ± 0.0002	dns_qclass
0.0007 ± 0.0001	weird_notice
0.0004 ± 0.0002	ssl_resumed
0.0003 ± 0.0001	ssl_established
0.0003 ± 0.0001	weird_addl
0.0003 ± 0.0001	http_version
0.0002 ± 0.0001	weird_name
0.0002 ± 0.0001	ssl_version
0.0001 ± 0.0000	http_uri
0.0001 ± 0.0000	src_bytes
... 18 more ...	

(f) ToN_IoT features extracted by PFI

Figure 3.4: Top 20 relevant features of attacks that binary classifiers learned from the three datasets (extracted by SHAP and PFI)

Table 3.4: Results about Binary Classifiers

Dataset	Accuracy	Precision	Recall	F1-score
NSL-KDD with SHAP features	0.787	0.970	0.646	0.776
NSL-KDD with PFI features	0.817	0.963	0.706	0.815
UNSW-NB15 with SHAP features	0.864	0.808	0.987	0.889
UNSW-NB15 with PFI features	0.859	0.801	0.990	0.885
ToN_IoT with SHAP features	0.872	0.790	0.866	0.826
ToN_IoT with PFI features	0.873	0.787	0.876	0.829

mutation importance without showing how much each features matters. This would lead to drawbacks in the case of features that have a large impact on a few predictions but no impact in general or features that have an average impact on all predictions.

SHAP presents the list of the most important features by plotting positions on the graphs shown in Figures 3.4a, 3.4c and 3.4e. We need to analyse each dot to understand the plot since each dot represents an instance in the dataset. The dot's vertical position indicates the feature it is depicting, its horizontal position refers to the impact of that value on the model's prediction. The colors of the dot represent the value of that feature for that instance of the dataset, indicating whether it is high, medium or low (red, purple or blue, respectively). For example, in Figure 3.4a, a high value of the feature *'Count'* increased the probability that the activity is attack by 20% to 60%. SHAP gives a different view of feature importance and shows how each features matters. Therefore, it mitigates the major disadvantage of PFI's explanations that we discussed above.

Evaluation indicators in Table 3.4 show that when the classifiers are trained on subsets of input features, their performance only changes slightly and even gets better in a few cases. Models that are trained on the PFI subset of UNSW-NB15 and NSL-KDD datasets get higher recall and accuracy, respectively. Models that are trained on the SHAP subset of ToN_IoT and NSL-KDD datasets get higher precision. This means that the sets of input features collected in these three datasets are not yet optimal to detect attack activities. Therefore, we combine the most important features extracted by SHAP and PFI (Figure 3.4) and use them to re-train the binary classifier to detect attacks without altering the model's parameters. Results are shown in Table

Table 3.5: Results of Binary Classifiers with Combined Set of Features

Dataset	Accuracy	Precision	Recall	F1-score
NSL-KDD with combined features	0.787	0.971	0.645	0.775
UNSW-NB15 with combined features	0.858	0.797	0.994	0.885
ToN_IoT with combined features	0.873	0.785	0.879	0.829

Table 3.6: Training time and detection time taken (using different set of features)

	Original	SHAP	PFI	Combined
NSL-KDD training time (ms)	88801	87006	87520	87815
Total detection time (ms)	507	434	448	470
UNSW-NB15 training time (ms)	128519	123377	122447	121950
Total detection time (ms)	1444	1268	1336	1292
ToN_IoT training time (ms)	257544	250684	252824	259633
Total detection time (ms)	1664	1458	1508	1493

3.5. Models that are trained on the combined subset of UNSW-NB15 and NSL-KDD datasets get higher recall and precision, respectively.

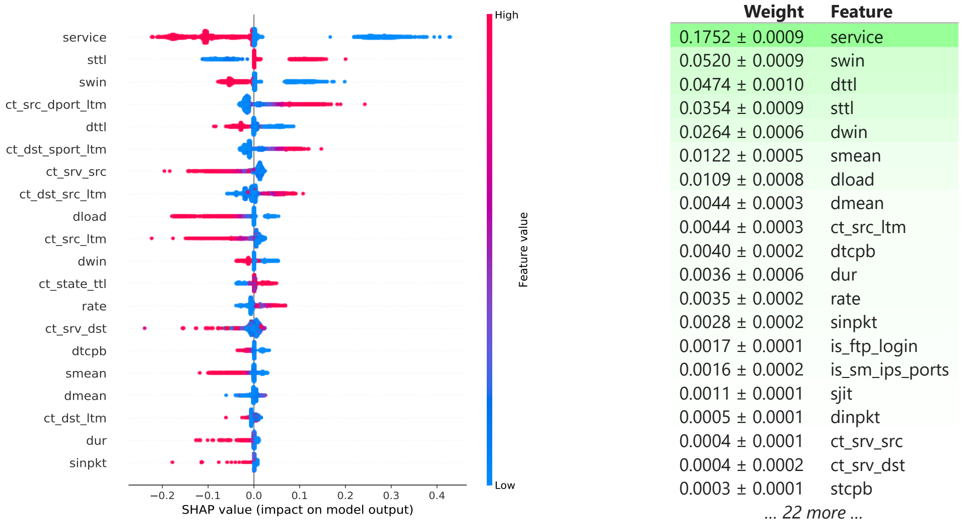
We also measure the time taken when using a different set of features for training and evaluating the models, as shown in Table 3.6. In all three datasets, using the set of features extracted by SHAP reduces both training time and detection time without significantly lower the model's performance. The training time tends to be reduced much more if its value is already high (when using the original features). Therefore, using SHAP features would reduce the network overhead and computational resources without lowering the effectiveness of IDS in IoT networks.

3.5.7 One-vs-All Classifiers

The next stage is to build one-vs-all classifiers to detect a specific class of attack and apply both global and local XAI methods. We expect that the explanations from the proposed framework would provide insight into the functioning of the models and help security experts in studying the characteristics of each cyber attack class. After building classifiers to detect individual attack classes, it is observed that the model can well detect a few attack classes, whilst the performance of classifiers in detecting

Table 3.7: Results about one-vs-all classifiers

Attack class	Dataset	Accuracy	Precision	Recall	F1-score
DoS	NSL-KDD	0.931	0.958	0.829	0.889
Probe	NSL-KDD	0.946	0.819	0.636	0.716
Exploits	UNSW-NB15	0.920	0.763	0.596	0.669
Generic	UNSW-NB15	0.991	0.999	0.962	0.980
Reconnaissance	UNSW-NB15	0.979	0.812	0.664	0.731
DDoS	ToN_IoT	0.987	0.822	0.902	0.859
XSS	ToN_IoT	0.969	0.629	0.736	0.678



Weight	Feature
0.1752 ± 0.0009	service
0.0520 ± 0.0009	swin
0.0474 ± 0.0010	dttl
0.0354 ± 0.0009	sttl
0.0264 ± 0.0006	dwin
0.0122 ± 0.0005	smean
0.0109 ± 0.0008	dload
0.0044 ± 0.0003	dmean
0.0044 ± 0.0003	ct_src_ltm
0.0040 ± 0.0002	dtcpb
0.0036 ± 0.0006	dur
0.0035 ± 0.0002	rate
0.0028 ± 0.0002	sinpkt
0.0017 ± 0.0001	is_ftp_login
0.0016 ± 0.0002	is_sm_ips_ports
0.0011 ± 0.0001	sjit
0.0005 ± 0.0001	dinpkt
0.0004 ± 0.0001	ct_srv_src
0.0004 ± 0.0002	ct_srv_dst
0.0003 ± 0.0001	stcpb

… 22 more …

(a) UNSW-NB15 features extracted by SHAP (b) UNSW-NB15 features extracted by PFI

Figure 3.5: Top 20 important features of *Generic* attack class that models have learned from UNSW-NB15

other classes is insufficient for applying the proposed framework. Results from training one-vs-all classifiers are shown in Table 3.7.

Among all classes of cyber attack in Table 3.7, the models can obtain the best performance on *Generic*. Therefore, we apply the XAI methods in analysing the model's decision in detecting *Generic*. Based on the explanations generated by both SHAP and PFI in Figure 3.5, the following features are considered as the most important features to detect a *Generic* attack:

- *service*: Low values of *service* feature would not affect the model's decision, but in many other cases, they increases the probability that the model identify the

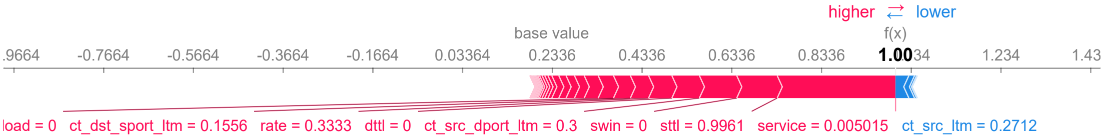


Figure 3.6: A local explanation on the *Generic* attack class

records as *Generic* attack by 25% - 40%. List of services in ascending order of value: *http, ftp, smtp, pop3, dns, snmp, ssl, dhcp, irc, radius, ssh, none.*

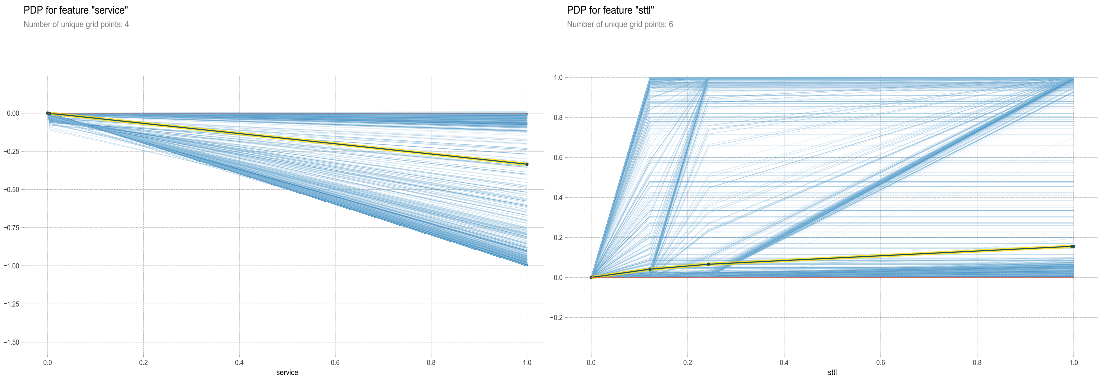
- *swin*: It refers to the values of source TCP window advertisement. Low values of *swin* feature would not affect the model's decision, but in many other cases, they increase the probability of a prediction as *Generic* attack by 10% - 20%.

- *dttl*: This feature measures the time to live of destination to source connections. Low values of *dttl* feature would slightly increase the chance that the record are detected as *Generic* attack.

- *sttl*: This feature measures the time to live of source to destination connections. In contrast to *dttl*, high values of *sttk* feature would increase the chance that the record is the *Generic* attack by 10% - 20%.

- *ct_src_dport_ltm*: This feature refers to the number of connections of the same source address and the destination port in 100 connections according to the last time. Interestingly, SHAP differs from PFI in computing the effect of this feature on the model's decisions. In this case, PFI hinders the feature's impact because PFI does not demonstrate how much this feature matters. PFI only computes the medium permutation importance and neglects data records in which this feature has a great impact.

SHAP can also be used to generate individual explanations. Figure 3.6 shows a data record that is classified as *Generic* attack. The model's decision is decomposed into the sum of effects of each feature value. In this specific data instance, $service = 0.005015$ (dns), $sttl = 0.9961$ (254) and $ct\_src\_dport\_ltm = 0.3$ (16) are the three features that contribute the most to the final decision of the model.



(a) feature *service*

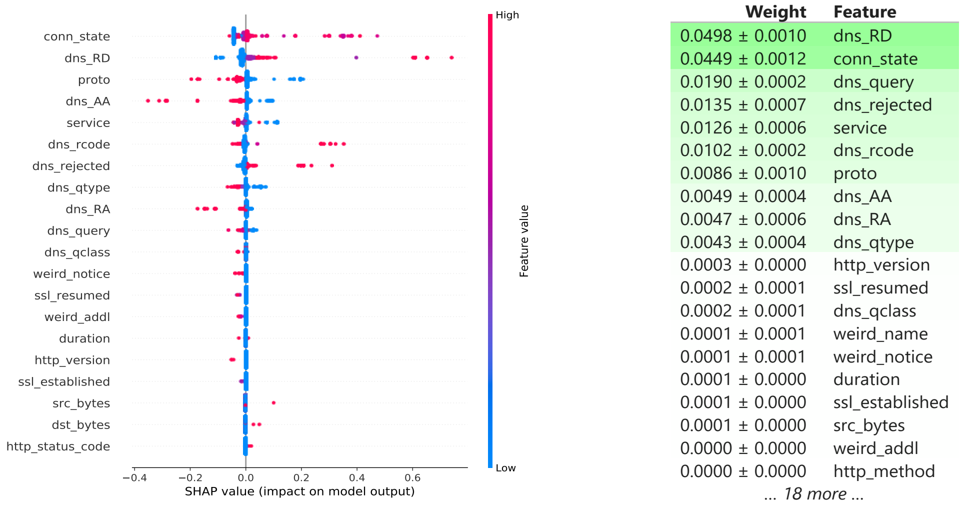
(b) feature *sttl*

Figure 3.7: PDP and ICE for features *service* and *sttl* feature in detecting *Generic* attack

To further analyse the effects of these features on the overall decision or individual decisions of the model, we apply PDP and ICE as shown in Figure 3.7a and 3.7a. ICE lines are blue and the average of all blue lines is yellow lines, also known as PDP. These methods demonstrate the model's prediction changes when a feature changes. Figure 3.7a tell us that the increase in value of *service* feature would lead to decrease in the probability that the model classify an activity as *Generic* attack. Moreover, it can be divided into two groups of instances depending on how much the value of *service* feature affects the model's prediction. One group slightly decreases the chance of a *Generic* attack by only 10%, whilst the other group significantly decreases such chance by 90% and up to 100%.

Figure 3.7b is more complex as the data instances can be divided into 4 groups. Data records in the first and second groups would be classified as *Generic* attack if the value of *sttl* feature is around 0.1 and 0.25, respectively. Whilst data records in the third group are classified as an attack if the value of *sttl* feature reaches its maximum, the effect of *sttl* feature on data instances' decisions in the fourth group is very small.

The one-vs-all classifier that detects *DDoS* attack in ToN_IoT dataset also perform well, which is shown in Table 3.7. This good performance allows us to further analyse the results by using XAI methods as we had used above to study *Generic* attack. Figure 3.11 show the list of the 20 most important features extracted by SHAP and PFI from



Weight	Feature
0.0498 ± 0.0010	dns_RD
0.0449 ± 0.0012	conn_state
0.0190 ± 0.0002	dns_query
0.0135 ± 0.0007	dns_rejected
0.0126 ± 0.0006	service
0.0102 ± 0.0002	dns_rcode
0.0086 ± 0.0010	proto
0.0049 ± 0.0004	dns_AA
0.0047 ± 0.0006	dns_RA
0.0043 ± 0.0004	dns_qtype
0.0003 ± 0.0000	http_version
0.0002 ± 0.0001	ssl_resumed
0.0002 ± 0.0001	dns_qclass
0.0001 ± 0.0001	weird_name
0.0001 ± 0.0001	weird_notice
0.0001 ± 0.0000	duration
0.0001 ± 0.0000	ssl_established
0.0001 ± 0.0000	src_bytes
0.0000 ± 0.0000	weird_addl
0.0000 ± 0.0000	http_method
… 18 more …	

(a) ToN_IoT features extracted by SHAP (b) ToN_IoT features extracted by PFI

Figure 3.8: Top 20 important features of *DDoS* attack class that models have learned from ToN_IoT

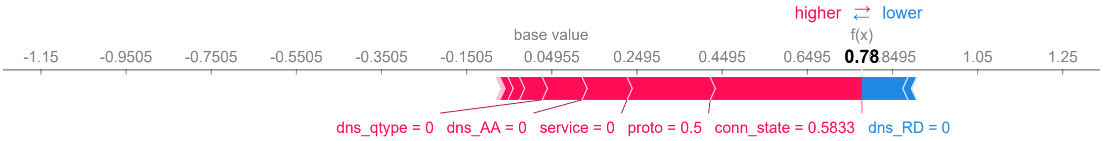
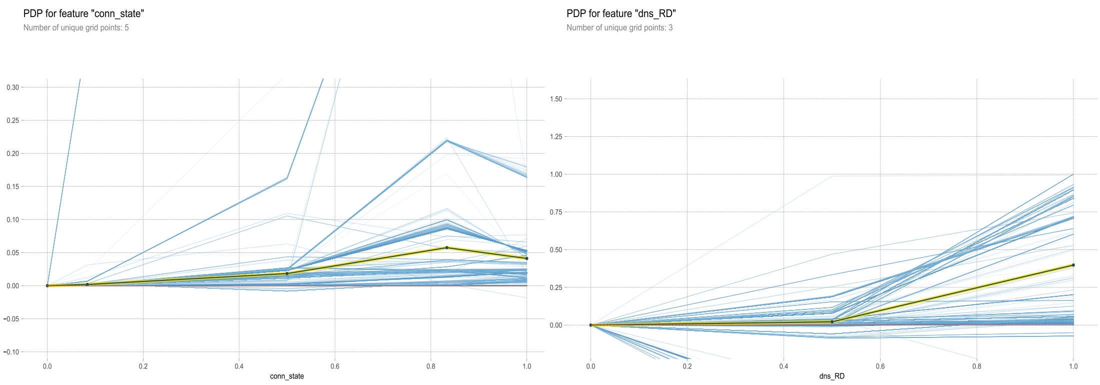


Figure 3.9: A local explanation on the *DDoS* attack class

the ToN_IoT dataset. Both SHAP and PFI extract some notable features, including:

- *conn_state*: this feature is a string that refers to the state of connections, such as S0 (connection without replay), S1 (connection established), and REJ (connection attempt rejected). Higher values of this feature would lead to higher probability that the data instances are classified as *DDoS* attacks. Symbols of *conn_state* feature in ascending order of value: *OTH, REJ, RSTO, RSTOS0, RSTR, RSTRH, S0, S1, S2, S3, SF, SH, SHR*

- *dns_RD*: this feature has boolean data type that indicates the recursion desire of DNS. After applying label encoding and scaler, the data pre-processor assigns value of "-" as 0, value of *False* as 1 and value of *True* as 2. Same as *conn_state* feature, higher value of *dns_RD* increase the chance that an instance is classifed as *DDoS* attack.



(a) feature *conn_state*

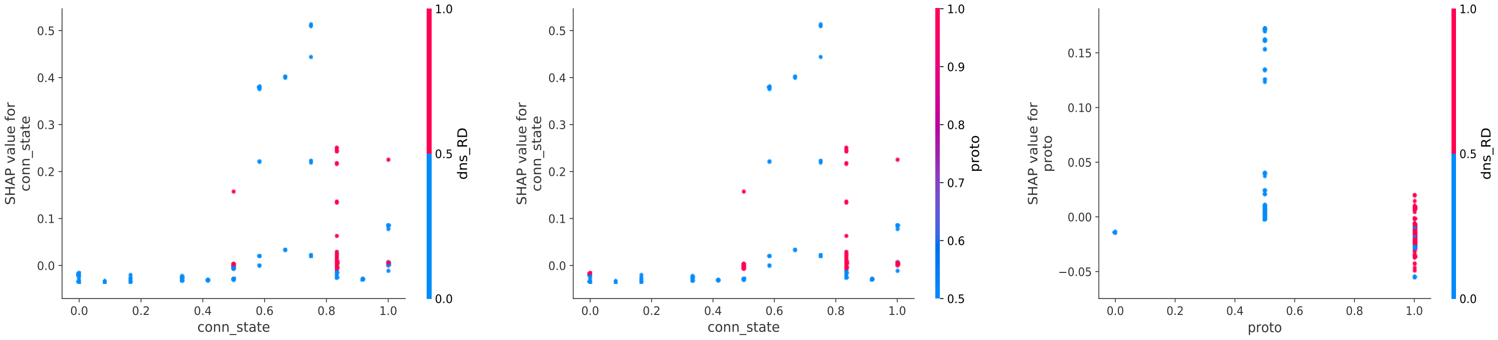
(b) feature *dns_RD*

Figure 3.10: PDP and ICE for features *conn_state* and *dns_RD* feature in detecting *DDoS* attack

Figure 3.9 demonstrates an individual explanation for a single data instance that is correctly classified as *DDoS* attack by the model. The value "0.5833" of feature *conn_state* contributes greatly to the final decision, which means that if *conn_state* equals to S1 the activity is likely classified as *DDoS* attack. Moreover, $proto = 0.5$ (TCP) and $service = 0$ (none) also contribute to such decision. Although $dns\_RD = 0$ (none) decreases the chance of a *DDoS* attack, the model has computed that overall probability equals to 0.78. Therefore, this is a *DDoS* attack.

Figure 3.10a and 3.10b shows PDP and ICE for the two features *conn_state* and *dns_RD*. Overall, high values of both of these features would increase the chance of a *DDoS* attack. Specifically, the feature *dns_RD* significantly increase such probability if it reaches maximum value. However, compared to Figure 3.7a and 3.7b, there are many data instances in Figure 3.10a and 3.10b that do not follow the rules mentioned above, which signifies hidden relationships between input features. This requires us to further analyse the results and demonstrate any existing correlations between features that can affect the final decision of the model.

Figure 3.11 demonstrates SHAP dependence plot for the three most important features consisting of *conn_state, dns_RD* and *proto.* Each dot on these plots represents a data instance in the dataset. The horizontal position indicates the value of the feature on the x-axis and the colour represents the value of other features shown on the right



(a) conn_state and dns_RD

(b) conn_state and proto

(c) proto and dns_RD

Figure 3.11: Dependence contribution plots of features *proto*, *dns_RD* and *conn_state*

column. Whilst the vertical position refers to the SHAP value that indicates how much these values of the two features affect the model's decision. Before analysing these plots, there are three different types of the protocol used in ToN_IoT, including ICMP (labeled as "0"), TCP (labeled as "0.5") and UDP (labeled as "1").

Figure 3.11a and 3.11b are quite similar in terms of distributions. Both plots observe high value of SHAP when the value of *conn_state* is in range of [0.5, 1]. Specifically, SHAP value is high when the value of *conn_state* is in range of [0.6, 0.8] and the values of *dns_RD* or *proto* are low. Whilst, high value of *dns_RD* or *proto* would significantly increase SHAP value when *conn_state*'s value is smaller than 0.6 or higher than 0.8. Figure 3.11c shows that all detected DDoS attacks use the UDP protocol.

3.5.8 Relationship Between Feature Values and Attacks

Applying local and global XAI methods on one-vs-all classifiers provides us with interesting results that reflect insight into the datasets and characteristics of each attack class. The relationships between the input features and individual class of attack are demonstrated through visualisation using our proposed framework. These explanations are intuitive and would be easy to understand for different types of XAI recipients with various levels of knowledge about cyber security or artificial intelligence. Therefore, the SPIP framework can effectively enhance the benefits that XAI models could bring to audiences.

The experiments in this chapter have shown a strong relationship between input features and attack classes. By applying the proposed framework on well-performing one-vs-all classifiers, we can analyse such relationships and the decision process of the AI-based models. For example, the framework studies that features *conn_state, dns_RD* and *proto* have strong relationships with *DDoS* attacks, which matches the real characteristics of *DDoS* attacks. This indicates that the classifier has given valid predictions as it has studied the right patterns of the particular attack class. Moreover, this has proven that the explanations generated by the SPIP framework can be used to effectively enhance the interpretability and explainability of AI-based intrusion detection models.

3.5.9 Discussion

There are several distinct and significant benefits to applying the proposed framework. These are:

- The proposed framework solves the problem of inaccessible explanations in achieving XAI. The outputs generated by SPIP framework are intuitive and easy to understand for any type of audience. By utilising visualisation-based XAI methods, the explanations can be widely adopted by society, especially policy-makers and the law.
- The framework assists developers and researchers in evaluating the quality, analysing and detecting any bias in the dataset. This is done by combining the foundation knowledge about individual attack classes and the explanations generated from the SPIP framework. More importantly, this framework can aid developers in designing more robust data-processing module by analysing the datasets and selecting the most informative input features. This reduces detection time of the model and may allows real-time detection of the IDS.
- SPIP's reasonable explanations support the interpretability of the IDS. It provides the recipient with the demonstration of relationships between features' value and the prediction as well as relationships between input features. Moreover, the framework is post-hoc, which can be applied to any model regardless of the underlying algorithms.
- SPIP evaluates the quality of the collected input features in detecting different attack classes. Each attack class has specific characteristics that are shown in the features' value. A quality set of input features would demonstrate these characteristics, hence enhancing trust in the models' decisions. Therefore, outputs from SPIP can help researchers in evaluating the input features and discover novel features to collect from network activities.

- Zero-day attacks exploit unknown vulnerabilities in the system; therefore, it is difficult to detect and analyse them. AI-based IDS can detect zero-day attacks and the SPIP framework can explain such decisions. SPIP's outputs would enhance the investigation process by helping security experts in discovering these unknown attacks' specific characteristics.

That said, there are also limitations to employing such a framework. The drawbacks of the SPIP framework:

- As outlined, strong IDS performance requires a comprehensive training set with valuable collected features from network activities. This is a significant challenge in building an effective IDS, as it is impossible to construct a dataset that involves all normal and malicious behaviours in a heterogeneous environment of IoT networks. Moreover, many existing datasets collect an incomplete set of features, and network information is captured without including both headers and payloads [14]. However, the SPIP framework can support security experts in collecting more relevant features from the network traffic, hence enhancing classifiers' performance.
- The generated explanations are limited to presenting notable features and their correlations without determining the security vulnerabilities that the attack exploits. However, security experts would utilise those results to discover and demonstrate the existing vulnerabilities based on their knowledge about the victim system and the notified attacks.

3.6 Conclusion & Future Work

This work has proposed the SPIP framework, developed to evaluate explainable DL algorithms for IDS deployments in IoT environments. The SPIP framework generates both local and global explanations and combines various XAI methods, including Shapley Additive Explanations (SHAP), Permutation Feature Importance (PFI), Individual

Conditional Expectation (ICE) and Partial Dependence Plot (PDP). The generated explanations by our framework are intuitive and easy for audiences of different technical levels to understand. The SPIP framework extracts a customised set of input features that outperforms the original set in the three datasets, including NSL-KDD, UNSW-NB15 and ToN_IoT. More importantly, the relationships extracted by SPIP matches the characteristics of specific attack classes. The framework proposed in this research would significantly enhance the utilisation of AI-based IDS in the cyber defence system.

SPIP has some limitations which would be improved in future research. This framework's outputs greatly rely on the performance of intrusion detection models, which means that the outcomes are not accurate when applying on an IDS with poor performance. Moreover, the proposed framework cannot identify the existing vulnerability that an attack class is exploiting. Future research can build a framework that can suggest potential vulnerabilities by studying the features of network traffic generated during cyber-attacks. The field of explainable AI has significant uses in cyber defence, and the development of IDS for IoT platforms is a critical area that would significantly benefit from this work. The SPIP framework seeks to begin to address this issue. Next chapter, An effective-flexible-explainable IDS is proposed to give more insights of the impact of explaining the data features of cyberattacks in IoT networks.

Chapter 4

An Effective, Explainable and Adaptable Intrusion Detection System in IoT networks

4.1 Introduction

Intrusion Detection System (IDS) is an effective solution to detect cyber-attacks, it has attracted the attention of experts to protect IoT networks [178]. IDS monitors and analyses network traffic and attempt to detect any cyber-attack based on sign of malicious behaviours. There are two main types of IDS, namely signature-based IDS and anomaly-based IDS. The anomaly-based is more advanced as it uses machine learning techniques to detect unknown attacks or attacks that use innovative techniques. This solution overcome some disadvantages of signature-based IDS, however, researchers would face numerous challenges to design an effective anomaly-based IDS, namely feature selection, diversity of detection algorithms and imbalanced datasets. This chapter aims to propose an architecture that tackles these problems.

A big problem with protecting IoT platforms is that IoT devices generate a massive amount of complex network data. It is estimated that the IoT is projected to reach 83 billion devices by 2024 [179]. Therefore, such a heterogeneous environment would generate high volume, speed, and variety of network data, which has made the data analytics process to build an effective IDS very difficult [180]. Anomaly-based IDS is trained on datasets that consisting of IoT network data, and their process would be adversely affected by redundant and irrelevant data features in these high-dimensional datasets [181]. Moreover, ML models that were trained by uninformative data features would be computationally expensive and likely to have low performance, hence increasing false alarm rate and making the process time-consuming [182,183].

Existing datasets that are used for building IDS that consists of numerous types of cyberattacks; however, an individual model that uses a specific method or algorithm may not perform well in detecting all attack classes [181]. Therefore, a practical model should combine different learning techniques to add diversity and increase the chance of detecting intrusions. Moreover, a combination of various classifiers makes the system more flexible and adaptable to heterogeneous IoT platforms.

ML-based IDSs' performance in real-life deployment heavily relies on the quality of the training dataset. Some datasets collect irrelevant data features or have an improper distribution of the number of attack classes. Training set with poor quality would make intrusion detection models less effective and adaptable to detect novel attacks [181]. Therefore, evaluating the quality of the training set is necessary. By doing so, researchers can analyse the advantages of each dataset and realise different patterns and data features of a specific attack class that models have learned while being trained on these datasets.

To tackle the problems of feature selection and diversity of detection algorithms, we propose an Effective-Flexible-Explainable (EFE) IDS. The proposed model is built upon the SPIP framework presented in Chapter 3, with further applications in selecting optimised set of features, and promoting flexibility for IDS. Other than interpreting predictions of ML classifiers, the model's goal is to design an adaptable IDS that is computationally inexpensive and time-saving in an ever-changing IoT environment.

Moreover, the EFE IDS can assist researchers in analysing imbalanced datasets and design an optimized datasets to train and evaluate models.

The remainder of this chapter is structured as follows. Section 4.2 discusses the related works. The structure of this model is described in Section 4.3, including the binary classifier and Target-vs-normal classifier. Section 4.4 presents the experiments and the results' discussions. Finally, Section 4.5 summarise this work and discusses the future research direction.

4.2 Related Work

In general, to tackle those problems mentioned above, researchers usually use two methods, including feature selection technique and hybrid algorithm to design an effective and flexible IDS. Authors apply different techniques to select the most informative features to decrease the training and processing time of the model. They also combine multiple algorithms or separate the training and detection of different attack classes to work with imbalanced datasets and improve the overall detection rate.

Alhakami et al. [178] proposed an IDS framework based on a Nonparametric Bayesian Approach and Feature Selection to enable the automatic removal of irrelevant features during the clustering process. This framework aims to prevent uninformative features from forming false clusters; thus, the IDS would generate less false alarms and overall process would be less time-consuming. The main advantages of the proposed framework are that it uses a flexible Bayesian statistical model to formalise the prior knowledge through probability [184] and an unsupervised feature selection mechanism to train the IDS effectively.

Thaseen et al. [185] proposed an Integrated IDS using Chi-Square Feature Selection and Ensemble of Classifiers. The data features in the training set are chosen based on Chi-square statistical significance test, and predictions are made by a voting mechanism including different classifiers instead of a single classifier.

Zong et al. [186] investigated a two-stage classifier approach to NIDS to address the problem of imbalanced training sets in constructing the ML-based IDS. The authors separate the processes of training and detection of different attack classes to improve the model's overall performance. This approach promotes flexibility in the classification process as different classifiers would be used for each stage of the NIDS. Moreover, over-sampling and under-sampling techniques were utilised for training the model effectively with imbalanced training sets.

Aljawarneh et al. [187] design a two-stage approach to build an effective IDS. The first stage is feature selection using a voting mechanism with Information Gain that combines the probability distributions of base learners. The second stage involves a hybrid algorithm consisting of multiple classifiers. The results obtained show improved accuracy and a low false alarm rate.

4.3 Structure of the EFE system

This section demonstrates the proposed model for an effective, flexible and explainable intrusion detection system. With the SPIP framework presented in Chapter 3, we are provided with reasonable explanations for each decision of the AI-based IDS. Furthermore, security experts would gain insight into the models' functioning as the SPIP framework can generate a list of relevant features for each attack class. We propose a model to build an effective, flexible and explainable IDS to enhance the collected results further.

Components in the proposed model can be categorized into two parts, including binary classifiers and target-vs-normal classifiers. The binary classifiers focus on detecting malicious activities regardless of the attack types or classes. Thus, security professionals can utilize these classifiers to build a baseline for normal behaviours in the network by applying the SPIP framework in Chapter 3. Meanwhile, the target-vs-normal classifiers aim to detect specific attack classes that they are trained and tasked to do. This classifier would distinguish between normal traffic's behaviours and a specific attack

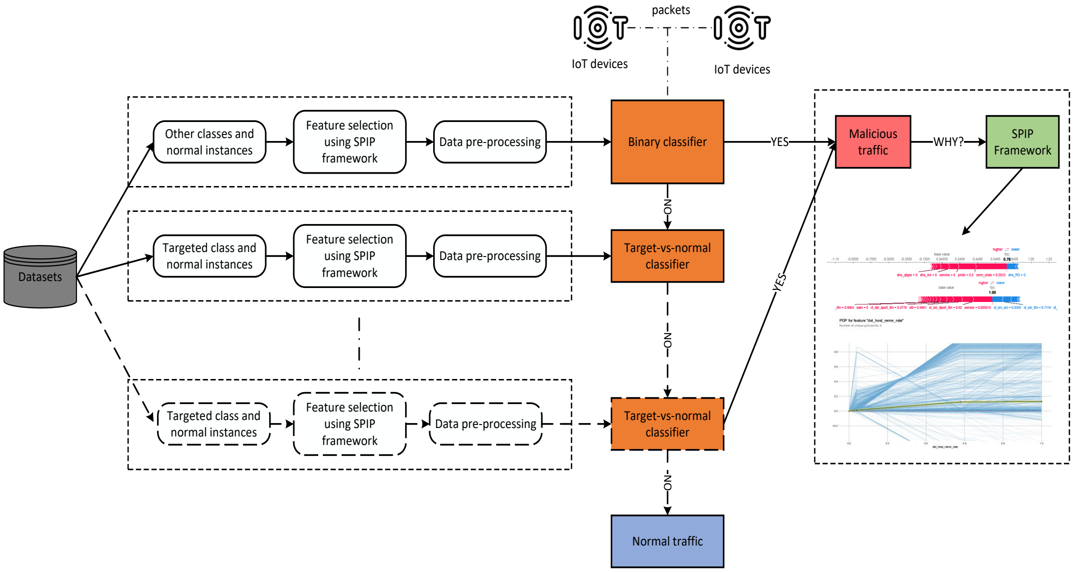

packets
IoT devices
IoT devices
Other classes and normal instances
Feature selection using SPIP framework
Data pre-processing
Binary classifier
YES
Malicious traffic
WHY?
SPIP Framework
NO
Datasets
Targeted class and normal instances
Feature selection using SPIP framework
Data pre-processing
Target-vs-normal classifier
YES
NO
Targeted class and normal instances
Feature selection using SPIP framework
Data pre-processing
Target-vs-normal classifier
NO
Normal traffic

Figure 4.1: The proposed architecture

class's characteristics. Furthermore, they can explain their decisions based on the input features they used for training the classifier.

4.3.1 Binary classifiers

We built binary classifiers upon Deep Neural Network (DNN), then trained and evaluated them using a customised version of NSL-KDD, UNSW-NB15 and ToN_IoT. The DNN is generally utilised to effectively analyse and detect patterns in large-scale data such as these benchmark datasets for IDS. Moreover, the SPIP framework would be applied to fitted binary classifiers to build a baseline for normal activities in the network. Specifically, the global explanations generated by SPIP can demonstrate a relationship between network features and malicious traffic. Furthermore, a set of relevant features for detecting attacks is also provided, which may be used to build more effective IDS or design better datasets. To build an effective and flexible IDS, we extract all data records for a few attack classes from the datasets, including NSL-KDD, UNSW-NB15, ToN-IoT. In other words, these binary classifiers would not be able to detect some attack classes. However, this disadvantage is compensated by target-vs-normal classifiers and their ability to identify and explain a specific malicious activity correctly.

4.3.2 Target-vs-normal classifiers

The target-vs-normal classifier only focuses on distinguishing between cyberattacks of a specific class and normal traffic. In this experiment, we built target-vs-normal classifiers upon DNN, then trained and evaluated them using a customised version of NSL-KDD, UNSW-NB15 and ToN_IoT. Data instances for normal activities are extracted from these datasets, together with records for a specific class depending on which class of cyberattack the classifier is tasked to detect. After target-vs-normal classifiers are trained, they would effectively detect the targeted attack classes and explain such predictions in real-time through the SPIP framework proposed in Chapter 3.

This component of the classifiers' system makes the proposed model flexible because target-vs-normal classifiers are ideal to be attached to any existing cyber defence system to detect a specific type of malicious activity and explain the predictions. They would be easily deployed when and where there is a need as well as being unequipped when and where there is no further need.

4.4 Experimentation and Results

To continue the work and apply the framework developed in Chapter 3, the experiment configurations, computer environment and evaluation metrics were not altered. In this section, we mainly demonstrate the data pre-processing for different types of classifiers and discuss the experiment results.

4.4.1 Datasets

The datasets used in Chapter 3 are continued to be used in this experiment, including NSL-KDD [174], UNSW-NB15 [27] and ToN_IoT [3]. All of these datasets contain contemporary normal and attack network traffics that were generated and collected in simulated IoT environments [176]. Therefore, they allow for practical training of AI-based IDS in IoT networks. These datasets were described in Chapter 3 and were utilised to evaluate the SPIP framework. Although we still use them in this Chapter, the difference lies in how they were pre-processed.

4.4.2 Data Pre-processing

Overall, we process input features using the same method as we did in Chapter 3. Symbolic features are converted using a label encoder, continuous features are normalized using a min-max scaler and binary features remain unchanged. More information regarding data pre-processing on input features is provided in Chapter 3 Section 3.5.

Table 3.7 in Chapter 3 has shown a list of attack classes that can be well detected by one-vs-all classifiers. In this chapter, we continue this work by building target-vs-normal classifiers for these attack classes using the list of relevant features extracted by the SPIP framework. The customized set of input features for each dataset are shown in Table 4.1.

4.4.2.1 Pre-processing for binary classifiers

With binary classifiers, we exclude data records of attack classes shown in Table 3.7, and we identify other classes in the three datasets as attack only, regardless of the class of cyberattack. To be more specific, *U2R* and *R2L* in NSL-KDD are labelled as *Attack*; *DoS (Denial of Service), Analysis, Fuzzers, Backdoor, Shellcode, Worms* in UNSW-NB15 are labeled as *Attack*; *DoS, Scanning, Backdoor, Injection, MITM (Man-in-the-middle), Password, Ransomware, XSS (Cross-site Scripting)* in ToN_IoT

Table 4.1: Customized set of feature for each dataset

Dataset	Features
NSL-KDD	dst_host_same_srv_rate, same_srve_rate, logged_in, rerror_rate, dst_host_rerror_rate, Count, dst_host_srv_count, dst_host_srv_rerror_rate, dst_host_same_src_port_rate, dst_host_serror_rate, dst_host_srv_serror_rate, protocol_type, dst_host_count, srv_serror_rate, service, srv_count, dst_host_diff_srv_rate, srv_diff_host_rate, srv_rerror_rate, diff_srv_rate, is_guest_login, serror_rate, num_shells, hot, root_shell
UNSW-NB15	sttl, ct_state_ttl, dttl, swin, dload, service, ct_dst_sport_ltm, ct_dst_src_ltm, ct_srv_dst, dwin, proto, dmean, stcpb, ct_srv_src, smean, rate, tcprtt, is_sm_ips_ports, synack, ct_src_dport_ltm, sbytes, ct_dst_ltm, spkts
ToN_IoT	proto, conn_state, dns_AA, dns_RD, dns_qtype, service, dns_rejected, dns_query, dns_rcode, dns_RA, dns_qclass, http_version, weird_notice, http_user_agent, weird_addl, ssl_resumed, ssl_established, http_status_code, http_uri, http_resp_mime_types, ssl_version, weird_name, src_bytes

are labeled as *Attack*. With the outcome being classified as *normal* or *attack*, models are trained using the customized set of input features shown in Table 4.1.

4.4.2.2 Pre-processing for target-vs-normal classifiers

Target-vs-normal classifiers aim to distinguish a specific class of attack from normal network traffic. Therefore, the number of classifiers depends on the number of attack classes in each dataset. Regarding the results obtained from one-vs-all classifiers from Chapter 3, seven target-vs-normal classifiers are constructed as one classifier per attack class, including *DoS* and *Probe* in NSL-KDD; *Exploits, Generic, Reconnaissance* in UNSW-NB15; *DDoS, XSS* in ToN_IoT. Each of these classifiers contains all data records of a specific attack class in the datasets, together with all data instances representing normal network activities. Therefore, the AI-based model would be more effective in analysing the unique patterns of each class of attack, highlighting its difference from normal network behaviours. Compared to the one-vs-all classifier, target-vs-normal is more advanced. It only focuses on characteristics of a specific attack class without considering relevant features of any other types of cyberattack.

Table 4.2: Results about Binary Classifiers

Dataset	Accuracy	Precision	Recall	F1
NSL-KDD	0.767	nan	0.0	nan
UNSW-NB15	0.699	0.437	0.982	0.605
ToN_IoT	0.881	0.781	0.815	0.798

Table 4.3: Results about Target-vs-normal Classifiers

Dataset	Accuracy	Precision	Recall	F1
DoS	0.8715	0.994	0.709	0.827
Probe	0.826	0.940	0.134	0.235
Exploits	0.962	0.900	0.941	0.920
Generic	0.992	0.989	0.987	0.987
Reconnaissance	0.975	0.829	0.896	0.861
DDoS	0.985	0.843	0.925	0.882
XSS	0.965	0.702	0.773	0.736

4.4.3 Results

We evaluate the binary classifiers, target-vs-normal classifiers, and the proposed EFE system and show the results in three separate table for comparison, including Table 4.2, 4.3 and 4.4, respectively. In general, the modified version of binary classifiers trained by the two datasets NSL-KDD and UNSW-NB15 perform poorly compared to the original version in Chapter 3. To compare between target-vs-normal in this Chapter and one-vs-all classifiers in Chapter 3, there are some attack classes in which target-vs-normal classifiers obtain higher performance, including *Exploits, Generic, Reconnaissance, DDoS* and *XSS*. Whilst evaluation metrics of the proposed EFE system is quite similar to Binary Classifiers with Combined Set of Features shown in Table 3.5 in the last Chapter, except for the NSL-KDD dataset in which the EFE system's performance is significantly lower.

Table 4.4: Results about the proposed EFE system

Dataset	Accuracy	Precision	Recall	F1
NSL-KDD	0.732	0.995	0.532	0.693
UNSW-NB15	0.814	0.749	0.995	0.855
ToN_IoT	0.871	0.791	0.860	0.824

4.4.4 Discussion

Based on the results shown in Table 4.2, 4.3 and 4.4, the modified version of binary classifiers have poor performance because the attack class' patterns in the three datasets complement each other, and the classifiers have to rely on the common characteristics of these attack classes to detect anomalies. Therefore, the absence of *DoS*, *Probe* in NSL-KDD or *Exploits, Generic, Reconnaissance* in UNSW-NB15 significantly decrease classifiers' capability to detect the leftover attack classes.

Based on the results in Table 4.2, the ToN_IoT dataset is less dependent on the common patterns of attack classes. The absence of a few attack classes does not affect the overall performance of the models. Moreover, attack classes in UNSW-NB15 and ToN_IoT have fewer characteristics in common, and the models have to analyse the unique characteristics of each class. Therefore, this promotes classifiers' ability to distinguish individual attack classes from normal behaviours. IN contrast, within NSL-KDD, attack classes have common patterns, and the model mainly relies on common characteristics to detect attacks. Therefore, the absence of *DoS* and *Probe* greatly degrade the model's capability to learn *U2R* and *R2L*'s characteristics, hence lowering the performance of the target-vs-normal classifiers.

The proposed intrusion detection model is constructed based on SPIP framework, thus it has advantages of the SPIP framework as outlined in Subsection 3.5.9 Chapter 3. More than that, it has the following unique advantages:

- Its explanations are more advanced as the model focus on distinguishing between a specific class of attack and normal behaviours. This allows researchers to analyse distinct characteristics of a specific class of attacks and their differences from other

Table 4.5: Training time and detection time taken of the proposed model

Dataset	Training time (ms)	Total detection time (ms)
NSL-KDD	66376	581
UNSW-NB15	84007	1857
ToN_IoT	193774	2001

attack types as well as benign network traffic. Moreover, this makes individual explanations of each alert more specific and accurate in the context of a particular type of attack.

- The proposed model is flexible as it is constructed from several removable components, including a binary classifier and different target-vs-normal classifiers. Suppose many attack classes no longer need to be detected. In that case, these removable components can be easily removed from the system, which would reduce network overhead and processing time for the cyber defence system. For example, if the web application in the network no longer uses any input from a user or the web developer has validated all input from the user, techniques like Cross-site Scripting (XSS) would not be effective and not be utilised by attackers. In such a case, IDS operators can remove the XSS target-vs-normal classifier from the IDS to prevent false-positive alerts.

- This architecture is constructed from various components, which enables the diversity of detection algorithms. This advantage can be utilised to build more robust detection model for each class of attack, hence mitigating the use of DL-based techniques which are computationally expensive. The extensive use of ML and DL-based techniques shows good performance, but can lead to latency issues [154]. To tackle this, less expensive detection algorithms can be combined in the proposed architecture and effectively detect cyber-attacks in IoT environment.

Similar to the SPIP framework, the proposed model's performance relies on the training set's quality with valuable features captured from network activities in IoT platforms. Based on the experiment results in this Chapter, we learn that the ToN_IoT dataset's customised set of input features are the most suitable for analysing distinct characteristics of each attack class because the binary classifier performs well with the absence of *DDoS* and *XSS*. In other words, the classifier was trained on a valuable set of features that highlights the unique characteristics of each attack class. In contrast, we observe a significant decrease in binary classifiers' performance with NSL-KDD and UNSW-NB15. This suggests that the set of input features in these datasets is insufficient in detecting different attack classes. Another drawback of the proposed IDS is the increase in detection time. Although the training time of the model is decreased, the more important factor - detection time increase significantly. More details are shown by comparing Table 4.5 and Table 3.6.

4.5 Conclusion & Future Work

This study has demonstrated a practical, flexible and explainable IDS in IoT networks. To tackle the problem of uninformative features in a training set of IDS, we utilise the SPIP framework to extract only relevant features related to specific attack classes, improving detection rate and reducing the process's time in target-vs-normal classifiers. The detection process of attacks combines multiple classifiers, which improves performance in some metrics and promotes the flexibility of the cyber defence system. Moreover, this allows different algorithms to be utilised to enhance the performance of each classifier instead of relying on an individual classifier. The underlying algorithm is not limited to DNN, as the SPIP framework can work with any model. Therefore, future research would aim to find an optimal algorithm to detect a specific attack class, improving overall performance. Another direction for future research would be building datasets that combine the most relevant features extracted from the three datasets used in this research. Then IDS are trained and evaluated to detect all attack classes in these datasets. In the next chapter, the conclusion and future research direction of this research have been described.

Chapter 5

Conclusion and Future Directions

5.1 Introduction

The rapid development and demand for IoT devices have introduced many vulnerabilities and novel attack vectors. Therefore, AI-based IDS with the capability to detect unknown attacks and perform well with large-scale complex data is in utmost need. However, AI techniques underpinning many cyber defence systems face a barrier that prevents its further application in critical sectors, which is the incapability of interpreting their decisions [73]. As such, the field of explainable AI-based IDS in IoT platforms must be addressed. This thesis examines the robustness of existing XAI methods, develops an explainable AI-based intrusion detection framework, and designs an adaptable and explainable intrusion detection architecture. In particular, the proposed IDS are explainable, adaptable and consume less time to be trained without a significant decrease in two training sets.

Chapter 2 has addressed the recent literature related to this research. The overview of broad topics including Cyber Defence, IDSs, AI and the IoTs have been discussed with a focus on existing XAI methods and the applications of AI in security for IoTs network environment. To be more specific, this chapter summarises and analyse recent work on XAI methods and ML-based IDS, hence discovering research opportunities to enhance existing systems by combining prominent approaches. Moreover, this chapter also discusses the research challenges of this thesis. In general, the Literature Review chapter has come to a conclusion that addresses the first sub-question and guides the direction for this research.

Next, Chapter 3 proposes an explainable intrusion detection framework named SPIP. This framework aims to interpret the predictions made by the decision engine of IDS. SPIP is a combination of various XAI methods that can generate both local and global explanations regardless of the underlying algorithms of the IDS. Moreover, this framework can analyse the training set and extract the most informative input features based on a theoretical foundation [110]. This advantage of SPIP reveals a research opportunity to design an effective and explainable IDS, which is presented later in Chapter 4. The conclusion of Chapter 3 summarises the benefits and drawbacks along with some future research directions to utilise or enhance the proposed framework.

In Chapter 4, we attempt to utilise the SPIP framework by designing an explainable and adaptable IDS in IoT networks. The proposed architecture aims to effectively detect ever-changing cyber-attacks in IoT platform by lowering the training time of the model, providing meaningful explanations, and being flexible. The performance of the model is compared with the models in 3 on the same datasets, and we observe a significant reduction in training time. However, the proposed model does not perform well with imbalanced datasets such as NSL-KDD, which has witnessed lower evaluation metrics.

The remainder of this chapter is structured as follows. Section 5.2 summarizes the research contribution and how this thesis addresses the research questions. Section 5.3 discusses the contribution to the knowledge of this work. Section 5.4 details methodological limitations and future research direction are discussed in section 5.5. Finally, the final remarks of this thesis are summarised in section 5.6.

5.2 Summary of Research

In Chapter 2 sub-section 2.4.3 and 2.6.3, the taxonomy of XAI and its necessity has been analyzed and discussed based on recent literature. A summary of various XAI methods has been detailed and evaluated. XAI approaches can be classified into intrinsic and post-hoc, and post-hoc methods are widely used due to their desirable advantages of model flexibility, explanation flexibility, and low switching cost. Next, further analysis of post-hoc methods has been detailed including each method's advantages and disadvantages in Table 2.1.

From the analysis of various XAI methods in Chapter 2, it was found that the relevance-based and visualisation post-hoc XAI methods are the most suitable to be utilised in cyber defence systems such as IDS. Other than the advantages of post-hoc approach, these XAI methods can generate meaningful and intuitive explanations. Relevance-based methods aim to compute an importance score of each input feature based on theoretical foundation, indicating how informative each data feature is and generate informative descriptions for IDS researchers and developers. Moreover, visualisation-based methods usually visualise the model's decisions by plotting graph, which effectively explains black-box models to various recipients who are not experts in AI techniques. As visualisation-based is a complex task, it is usually coupled with relevance-based methods to provide comprehensive explanations to the XAI audience [73].

The aim of combining various XAI methods is to overcome several challenges to achieving XAI listed in subsection 2.7.2, including accessible explanations and feature dependence. Although our main focus is the XAI framework, the IDS must also achieve a good performance so that the generated explanations can be meaningful and reflect the correct patterns of cyber-attacks. To do this, an IDS has been built upon Deep Neural Network (DNN), then being trained and evaluated by contemporary IoT network datasets, including NSL-KDD, UNSW-NB15 and ToN_IoT. The highest possible performance has been obtained through adjusting hyperparameters of the deep learning structure, including the optimizer, number of hidden layers, number of neurons in each hidden layer, training epoch, dropout rate and learning rate.

After finding the optimal set of parameters for the model, the proposed framework has been used to interpret the black-box model's predictions by giving both global and local explanations. The global explanations are generated by SHAP [98], PFI [97] and PDP [83], including a set of the most informative input features that the model has learned from each training set and a plot showing the relationship between a feature and model's decisions. Then, results obtained from these global explanations have been evaluated as we use them to re-train the binary classifier without altering the model's parameters. Experimental results show that models trained on the combined subset of UNSW-NB15 and NSL-KDD datasets get higher recall and precision, respectively. Moreover, PDP visualizes the relationship between a targeted feature and model's decisions, which is easy to understand for lay-user of the IDS.

The local explanations have been generated by SHAP [98] and ICE, which aims to interpret individual data instances through visualization. SHAP's plot presents each input feature's impact on the final decision by generating blocks of colour (red or blue). This method is intuitive, highlighting only a few most relevant features in a pleasant way that the users can easily comprehend. Instead of showing the effects of a set of features, ICE's plot aims to demonstrate a feature's impact on the decision of the model by drawing a line indicating how a decision would be made with a specific value of the targeted feature. Moreover, the framework enables security experts to discover hidden correlations between features that adversely affect the generated explanations. This is an excellent advantage of the framework as it can overcome the challenge of feature dependence when applying XAI methods.

We rely on the framework proposed in chapter 3 to design an effective and explainable IDS in an IoT network environment. The framework can promote the interpretability of any models underlying the IDS by its combined XAI methods. Moreover, the global explanations can extract a set of the most informative input features to train the ML model effectively, which has been shown in section 3.5. To further enhance the model's effectiveness, the training of the model to detect some classes of attack are separated, which allows us to adjust the model parameters and even the underlying algorithms that achieve the best performance. Moreover, this makes the cyber defence system more flexible as it is constructed from several removable components. Experimental results from chapter 4 show a significant decrease in performance of the proposed IDS with the NSL-KDD dataset. This reveals that this architecture does not perform well with imbalanced datasets where features of the significant attack classes play a prominent role in the training set. However, the proposed architecture has reduced training time and generate more advanced explanations to analyze the distinct characteristics of the various attack classes in an IoT network.

5.3 Contribution to Knowledge

The key contributions of this work are as follows:

- **In-depth analysis of existing XAI methods** that classifies different approaches and discusses XAI methods' advantages and disadvantages
- **Explainable Deep Learning-enabled Intrusion Detection Framework** combines the most prestigious methods that gives both global and local explanations for the cyber defence systems regardless of the underpinning decision engine's algorithms.
- **Evaluation of the XAI framework in IDS** is conducted through a data-driven problem-oriented methodological approach that show the advantages and disadvantages of the framework's impact on cyber defence system.
- **Effective, explainable and adaptable IDS** that perform well with ever-changing cyber-attacks in IoT environment and being able to interpret its own decisions based on a valid theoretical foundation.

5.4 Methodological Limitations

Limitations of the proposed framework and architecture throughout this thesis are discussed below:

Firstly, the proposed framework dramatically relies on the performance of the Intrusion Detection models. In other words, the generated outputs are not accurate when applying to an IDS with poor performance. Therefore, the limitations of this framework also cover the challenges to building an effective IDS, especially the requirement of a comprehensive training set. Fortunately, the architecture designed in chapter 4 can support security experts in evaluating the quality of datasets and collect more relevant features from the network traffic to enhance to classifiers' performance.

Secondly, the framework's explanations are limited to the demonstration of notable features, their correlations and their relationship with cyberattacks without identifying the security vulnerabilities that have been exploited. Therefore, the root causes of cyber-attacks are still unclear to XAI users. However, this provides an opportunity for researchers to utilise the outputs of the SPIP framework to discover and demonstrate the vulnerabilities based on their knowledge about the victim system and the notified cyber-attacks.

Lastly, the architecture proposed in chapter 4 does not perform well with imbalanced datasets. More specifically, the ML models only learn characteristics of major classes instead of common characteristics of every cyber-attack type presented in those datasets. Therefore, separating the training to detect major classes and other minor classes would result in poor performance of the model. We need to optimise the feature collection when building the datasets and store an equal number of instances representing each attack class to mitigate this.

5.5 Future Work

The following research direction will further improve the research conducted in this thesis

- **Development of Optimised Architectures for Adaptable and Explainable Intrusion Detection System -** Building on the work conducted throughout Chapter 4, an optimisation methodology can be implemented on the decision engine of classifiers in the architecture. The proposed architecture promotes adaptability, allowing individual classifiers to utilise different algorithms that fit each type of cyber-attack. Moreover, the SPIP framework can interpret the model's decisions regardless of the underlying algorithms, and it also benefits from the high performance of the classifier.

- **Development of datasets with optimised feature collection -** The set of input features plays an important role in effectively detecting a specific attack class. As shown in chapter 4, ToN_IoT dataset's customised set of input features are the most suitable for analysing distinct characteristics of attack classes, especially *DDoS* and *XSS* attacks. A promising research direction would be to combine the most informative features from different datasets and build a more comprehensive and effective dataset to train and evaluate intrusion detection models.
- **Development of model that can detect Multi-Stage Attacks** - Defending against Multi-Stage Attacks (MSA) remains a challenge for existing IDS techniques. IDS effectively detects single-stage attacks as the malicious activities are conducted over a short period. The MSA utilizes more complex attack techniques over a long period [30] in which each individual step taken is insufficient to be recognized as an aggression. Therefore, the MSA avoids being detected by IDS. In [188], the authors point out several challenges for detecting MSA, including: (1) modelling MSAs, (2) building a system to detect and track the progress of interleaved MSA, and (3) constructing datasets with interleaved MSA scenarios.
- **Standardised terminology and evaluation of explanations**: the XAI research community does not have a standardised terminology. Terminologies in XAI is defined differently by researchers as there are no standardised definitions and vocabularies [73], which would lead to confusion. This makes it impossible to evaluate and compare the performance of different XAI methods as the ground truth is unknown. Moreover, no criteria have been standardised that consider the subjective measures used in human-centred evaluations. It is also impossible to evaluate all XAI methods by a defined evaluation metric [16]. To develop a prac-

tical evaluation of XAI explanations, all the above aspects need to be considered carefully.

- **Improving statistical certainty in XAI methods**: Many XAI methods (PFI, SHAP) are subject to uncertainty as they provide explanations by computing from data. However, the uncertainty of the explanation is not addressed as explanations are given. Consequently, the explanations are not reliable and compromise the responsible AI. To fix this, a rigorous approach should be adopted to study the uncertainty of XAI methods [189]. Otherwise, XAI has to face statistical testing problems such as p-hacking [190].

- **Enhancing systematic stability of XAI methods**: System instability affects the performance of XAI methods that analyse the internal structure of the models. Many different ML models can perform well on a specifically labelled dataset; however, their internal pathway might differ due to the complexity of ML techniques. Such differences would lead to changes in generated explanations across multiple models [69].

- **Mitigating feature dependence**: Many XAI methods (PDP, LIME, PFI) suffer from the assumption of independence; thus, they automatically create invalid data points. Using such data points probably degrades the reliability of XAI methods as explanations are given unreal data instances and will never happen in real-life [189]. Technically, when features in the dataset are correlated, the explanations would be misleading.

5.6 Final Remarks

This thesis aims to address the questions *'How can we design an explainable Artificial Intelligence-enabled Intrusion Detection System that is effective and adaptable to ever-changing Internet of Things network environment?'* This research questions emerges from the necessity to explain Machine Learning models' decisions used in Intrusion Detection System in Internet of Things network environment. The heterogeneous environment of Internet-connected devices can enable various novel attack vectors and generate a high volume of complex network traffic, which requires the use of Artificial Intelligence to analyse patterns of cyber-attacks and detect zero-day attacks. However, Machine Learning models are black-box models, preventing ML techniques' development in further practical implementation. Therefore, there is a requirement to promote the interpretability of such models, analysing and explaining their decisions to various types of audiences and indirectly benefit the performance of the Cyber Defence system.

This thesis contributes to the field of Explainable Artificial Intelligence in Cyber Defence systems by; 1) developing an Explainable Intrusion Detection framework; and 2) designing an explainable, effective and adaptable intrusion detection architecture in IoT network environment. The ultimate goal of the research is to improve the interpretability of black-box models used in the Intrusion Detection System and evaluate the Explainable Artificial Intelligence's impact on designing an effective Cyber Defence system.

In conclusion, the proposed framework and architecture in this research provide a foundation for further work on Explainable Artificial Intelligence in Cyber Defence Systems. In the future, this work will assist researchers in utilising explainable AI for developing context-aware anomaly detection methods in IoT networks.

Bibliography

[1] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/7/1177

[2] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[3] N. Moustafa, "A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets," *Sustainable Cities and Society*, vol. 72, p. 102994, 2021.

[4] I. A. Khan, N. Moustafa, D. Pi, Y. Hussain, and N. A. Khan, "Dff-sc4n: A deep federated defence framework for protecting supply chain 4.0 networks," *IEEE Transactions on Industrial Informatics*, 2021.

[5] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on software engineering*, no. 2, pp. 222–232, 1987.

[6] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

[7] I. A. Khan, N. Moustafa, I. Razzak, M. Tanveer, D. Pi, Y. Pan, and B. Ali, "Xsru-iomt: Explainable simple recurrent units for threat detection in internet of medical things networks," *Future Generation Computer Systems*, 2021.

[8] S. Sarkar, T. Weyde, A. Garcez, G. G. Slabaugh, S. Dragicevic, and C. Percy, "Accuracy and interpretability trade-offs in machine learning applied to safer gambling," in *CEUR Workshop Proceedings*, vol. 1773. CEUR Workshop Proceedings, 2016.

[9] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[10] J. Navarro, A. Deruyver, and P. Parrend, "A systematic survey on multi-step attack detection," *Computers Security*, vol. 76, pp. 214–249, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818302141

[11] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. J. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, pp. 349–359, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18307532

[12] I. H. Sarker, M. H. Furhad, and R. Nowrozy, "Ai-driven cybersecurity: an overview, security intelligence modeling and research directions," *SN Computer Science*, vol. 2, no. 3, pp. 1–18, 2021.

[13] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of information security and applications*, vol. 44, pp. 80–88, 2019.

[14] N. Moustafa, J. Hu, and J. Slay, "A holistic review of Network Anomaly Detection Systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33–55, 2 2019.

[15] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, p. 68–77, Dec. 2019. [Online]. Available: https://doi.org/10.1145/3359786

[16] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/5/593

[17] S. Morgan, "Cybercrime to cost the world $10.5 trillion annually by 2025," *Cybercrime Magazine*, vol. 13, 2020.

[18] ACSC, "Australian cyber security centre annual cyber threat report 2020-21," Tech. Rep., 2021. [Online]. Available: https://www.cyber.gov.au/acsc/view-all-content/reports-and-statistics/acsc-annual-cyber-threat-report-2020-21

[19] E. G, "Us government to spend over $18 billion on cybersecurity - atlas vpn," 2020. [Online]. Available: https://atlasvpn.com/blog/us-government-to-spend-over-18-billion-on-cybersecurity

[20] B. von Solms and R. von Solms, "Cybersecurity and information security – what goes where?" *Information and Computer Security*, vol. 26, no. 1, pp. 2–9, 2018.

[21] H. Bing, S. Hao, Z. Fangwei, Z. Shuai, Q. Tao, and Y. Changjiang, "Application process of machine learning in cyberspace security," in *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*, 2021, pp. 865–869.

[22] D. J. Gunkel, *Hacking cyberspace.* Routledge, Abingdon, 2018.

[23] J. V. D. Ham, "Toward a better understanding of "cybersecurity"," *Digital Threats: Research and Practice*, vol. 2, no. 3, Jun. 2021. [Online]. Available: https://doi.org/10.1145/3442445

[24] J. B. Ulven and G. Wangen, "A systematic review of cybersecurity risks in higher education," *Future Internet*, vol. 13, no. 2, p. 39, 2021.

[25] S. Mittal, P. K. Das, V. Mulwad, A. Joshi, and T. Finin, "CyberTwitter: Using Twitter to generate alerts for cybersecurity threats and vulnerabilities," in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016.* Institute of Electrical and Electronics Engineers Inc., 11 2016, pp. 860–867.

[26] A. Bécue, I. Praça, and J. Gama, "Artificial intelligence, cyber-threats and industry 4.0: Challenges and opportunities," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3849–3886, 2021.

[27] N. Moustafa and 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)*, 2015, pp. 1–6.

[28] B. Alhayani, S. T. Abbas, D. Z. Khutar, and H. J. Mohammed, "Best ways computation intelligent of face cyber attacks," *Materials Today: Proceedings*, 2021.

[29] J. Zhao, S. Shetty, J. W. Pan, C. Kamhoua, and K. Kwiat, "Transfer learning for detecting unknown network attacks," *EURASIP Journal on Information Security*, vol. 2019, no. 1, pp. 1–13, 2019.

[30] J. Shin, S.-H. Choi, P. Liu, and Y.-H. Choi, "Unsupervised multi-stage attack detection framework without details on single-stage attacks," *Future Generation Computer Systems*, vol. 100, pp. 811–825, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18329212

[31] [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[32] I. Ghafir, V. Prenosil *et al.*, "Advanced persistent threat attack detection: an overview," *Int J Adv Comput Netw Secur*, vol. 4, no. 4, p. 5054, 2014.

[33] R. Bace and P. Mell, "Intrusion Detection Systems," National Institute of Standards and Technology (NIST), Tech. Rep., 2001.

[34] R. Wazirali, "An improved intrusion detection system based on knn hyperparameter tuning and cross-validation," *Arabian Journal for Science and Engineering*, vol. 45, no. 12, pp. 10 859–10 873, 2020.

[35] P. Sharma, J. Sengupta, and P. Suri, "Survey of intrusion detection techniques and architectures in cloud computing," *International Journal of High Performance Computing and Networking*, vol. 13, no. 2, pp. 184–198, 2019.

[36] S. A. Ludwig, "Intrusion detection of multiple attack classes using a deep neural net ensemble," in *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings*, vol. 2018-Janua. Institute of Electrical and Electronics Engineers Inc., 2 2018, pp. 1–7.

[37] D. Moon, S. B. Pan, and I. Kim, "Host-based intrusion detection system for secure human-centric computing," *The Journal of Supercomputing*, vol. 72, no. 7, pp. 2520–2536, 2016.

[38] H. A. Kholidy and F. Baiardi, "Cids: A framework for intrusion detection in cloud systems," in *2012 Ninth International Conference on Information Technology - New Generations*, 2012, pp. 379–385.

[39] L. Santos, C. Rabadao, and R. Gonçalves, "Intrusion detection systems in internet of things: A literature review," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018, pp. 1–7.

[40] G. De Carvalho Bertoli, L. A. Pereira Júnior, O. Saotome, A. L. Dos Santos, F. A. N. Verri, C. A. C. Marcondes, S. Barbieri, M. S. Rodrigues, and J. M. Parente De Oliveira, "An end-to-end framework for machine learning-based network intrusion detection system," *IEEE Access*, vol. 9, pp. 106 790–106 805, 2021.

[41] J. Peng, K.-K. R. Choo, and H. Ashman, "User profiling in intrusion detection: A review," *Journal of Network and Computer Applications*, vol. 72, pp. 14–27, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804516301412

[42] M. Kumar, M. Hanumanthappa, and T. V. Suresh Kumar, “Encrypted traffic and ipsec challenges for intrusion detection system,” in *Proceedings of International Conference on Advances in Computing*, A. Kumar M., S. R., and T. V. S. Kumar, Eds. New Delhi: Springer India, 2012, pp. 721–727.

[43] E. Besharati, M. Naderan, and E. Namjoo, “Lr-hids: logistic regression host-based intrusion detection system for cloud environments,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3669–3692, 2019.

[44] M. A. Rahman, A. T. Asyhari, L. Leong, G. Satrya, M. H. Tao, and M. Zolkipli, “Scalable machine learning-based intrusion detection system for iot-enabled smart cities,” *Sustainable Cities and Society*, vol. 61, p. 102324, 2020.

[45] M. F. Elrawy, A. I. Awad, and H. F. Hamed, “Intrusion detection systems for iot-based smart environments: a survey,” *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1–20, 2018.

[46] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in internet of things,” *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[47] J. Ribeiro, F. B. Saghezchi, G. Mantas, J. Rodriguez, and R. A. Abd-Alhameed, “Hidroid: Prototyping a behavioral host-based intrusion detection and prevention system for android,” *IEEE Access*, vol. 8, pp. 23 154–23 168, 2020.

[48] J. Ribeiro, F. B. Saghezchi, G. Mantas, J. Rodriguez, S. J. Shepherd, and R. A. Abd-Alhameed, “An autonomous host-based intrusion detection system for android mobile devices,” *Mobile Networks and Applications*, vol. 25, no. 1, pp. 164–172, 2020.

[49] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, “Intrusion detection system: A comprehensive review,” *Elsevier*, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2012.09.004

[50] N. Moustafa, G. Misra, and J. Slay, “Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web

application attacks," *IEEE Transactions on Sustainable Computing*, pp. 1–1, 2018.

[51] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. Khan, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," in *Procedia Computer Science*, vol. 171. Elsevier B.V., 2020, pp. 1251–1260. [Online]. Available: www.sciencedirect.com

[52] S. Omid Azarkasb, S. Sedighian Kashi, and S. Hossein Khasteh, "A network intrusion detection approach at the edge of fog," in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, 2021, pp. 1–6.

[53] M. Mulyanto, M. Faisal, S. W. Prakosa, and J.-S. Leu, "Effectiveness of focal loss for minority classification in network intrusion detection systems," *Symmetry*, vol. 13, no. 1, p. 4, 2021.

[54] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the cicids2017 case study," in *2021 IEEE Security and Privacy Workshops (SPW)*, 2021, pp. 7–12.

[55] N. Moustafa, G. Creech, and J. Slay, *Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models*. Cham: Springer International Publishing, 2017, pp. 127–156. [Online]. Available: https://doi.org/10.1007/978-3-319-59439-2_5

[56] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," vol. 46, no. 4, 2014. [Online]. Available: https://doi.org/10.1145/2542049

[57] D. Li, L. Deng, M. Lee, and H. Wang, "Iot data feature extraction and intrusion detection system for smart cities based on deep migration learning," *International journal of information management*, vol. 49, pp. 533–545, 2019.

[58] H. S. Anderson, A. Kharkar, B. Filar, and P. Roth, "Evading machine learning malware detection," *Black Hat*, 2017.

[59] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," *arXiv preprint arXiv:1710.00811*, 2017.

[60] Q. Chen, Q. Liao, Z. L. Jiang, J. Fang, S. Yiu, G. Xi, R. Li, Z. Yi, X. Wang, L. C. Hui *et al.*, "File fragment classification using grayscale image conversion and deep learning in digital forensics," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 140–147.

[61] X. Wang, J. Zhang, A. Zhang, and J. Ren, "Tkrd: Trusted kernel rootkit detection for cybersecurity of vms based on machine learning and memory forensic analysis," *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 2650–2667, 2019.

[62] X.-D. Zhang, "Machine learning," in *A Matrix Algebra Approach to Artificial Intelligence*. Springer, 2020, pp. 223–440.

[63] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, "Unsupervised learning based on artificial neural network: A review," in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, 2018, pp. 322–327.

[64] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[65] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[66] S. Raschka and V. Mirjalili, *Python Machine Learning - Second Edition*. Packt Publishing, 2017.

[67] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," *arXiv*, 1 2017. [Online]. Available: https://arxiv.org/abs/1701.02145v1

[68] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv*, 1 2019. [Online]. Available: https://arxiv.org/abs/1901.03407v2

[69] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (xai),” *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[70] J. D. Moore and W. R. Swartout, “Explanation in expert systemss: A survey,” University of Southern California Marina Del Rey Information Sciences Inst, Tech. Rep., 1988.

[71] V. Dignum, “Responsible artificial intelligence: Designing ai for human values,” 2017.

[72] S. R. Islam, W. Eberle, S. K. Ghafoor, and M. Ahmed, “Explainable Artificial Intelligence Approaches: A Survey,” *arXiv*, 1 2021. [Online]. Available: http://arxiv.org/abs/2101.09429

[73] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, “Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai,” *Information Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[74] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[75] M. T. Ribeiro, S. Singh, and 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.

[76] B. Kim, E. Glassman, B. Johnson, and J. Shah, “ibcm: Interactive bayesian case model empowering humans via intuitive interaction,” 2015.

[77] B. Mahbooba, M. Timilsina, R. Sahal, and M. Serrano, “Explainable artificial intelligence (xai) to enhance trust management in intrusion detection systems using decision tree model,” *Complexity*, vol. 2021, 2021.

[78] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (xai): A survey," *ArXiv*, vol. abs/2006.11371, 2020.

[79] B. Letham, C. Rudin, T. H. McCormick, D. Madigan *et al.*, "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model," *Annals of Applied Statistics*, vol. 9, no. 3, pp. 1350–1371, 2015.

[80] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning.* PMLR, 2015, pp. 2048–2057.

[81] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Machine Learning*, vol. 102, no. 3, pp. 349–391, 2016.

[82] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[83] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[84] Q. Zhao and T. Hastie, "Causal interpretations of black-box models," *Journal of Business & Economic Statistics*, vol. 39, no. 1, pp. 272–281, 2021.

[85] B. M. Greenwell, "pdp: An r package for constructing partial dependence plots." *R J.*, vol. 9, no. 1, p. 421, 2017.

[86] D. Apley, "Visualizing the effects of predictor variables in black box supervised learning models. arxiv," *arXiv preprint arXiv:1612.08468*, 2016.

[87] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015.

[88] A. Goldstein, A. Kapelner, and J. Bleich, "Icebox: Individual conditional expectation plot toolbox," 2017.

[89] N. Puri, P. Gupta, P. Agarwal, S. Verma, and B. Krishnamurthy, “Magix: Model agnostic globally interpretable explanations,” *arXiv preprint arXiv:1706.07160*, 2017.

[90] Y. Ming, H. Qu, and E. Bertini, “Rulematrix: Visualizing and understanding classifiers with rules,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 1, pp. 342–352, 2018.

[91] N. Frosst and G. Hinton, “Distilling a neural network into a soft decision tree,” *arXiv preprint arXiv:1711.09784*, 2017.

[92] L. Hu, J. Chen, V. N. Nair, and A. Sudjianto, “Locally interpretable models and effects based on supervised partitioning (lime-sup),” *arXiv preprint arXiv:1806.00663*, 2018.

[93] J. Rabold, H. Deininger, M. Siebers, and U. Schmid, “Enriching visual with verbal explanations for relational concepts–combining lime with aleph,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 180–192.

[94] A. H. A. Rahnama and H. Boström, “A study of data and label shift in the lime framework,” *arXiv preprint arXiv:1910.14421*, 2019.

[95] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[96] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[97] A. Fisher, C. Rudin, and F. Dominici, “All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously.” *Journal of Machine Learning Research*, vol. 20, no. 177, pp. 1–81, 2019.

[98] S. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *arXiv preprint arXiv:1705.07874*, 2017.

[99] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[100] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[101] J. Bien and R. Tibshirani, "Prototype selection for interpretable classification," *The Annals of Applied Statistics*, pp. 2403–2424, 2011.

[102] K. S. Gurumoorthy, A. Dhurandhar, G. Cecchi, and C. Aggarwal, "Efficient data representation by selecting prototypes with importance weights," in *2019 IEEE International Conference on Data Mining (ICDM).* IEEE, 2019, pp. 260–269.

[103] B. Kim, R. Khanna, and O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 2288–2296.

[104] P. Cortez and M. J. Embrechts, "Using sensitivity analysis and visualization techniques to open black box data mining models," *Information Sciences*, vol. 225, pp. 1–17, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025512007098

[105] P. Cortez and M. J. Embrechts, "Opening black box data mining models using sensitivity analysis," in *2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, 2011, pp. 341–348.

[106] A. Stojić, N. Stanić, G. Vuković, S. Stanišić, M. Perišić, A. Šoštarić, and L. Lazić, "Explainable extreme gradient boosting tree-based prediction of toluene, ethylbenzene and xylene wet deposition," *Science of The Total Environment*, vol. 653, pp. 140–147, 2019.

[107] M. Kłosok, M. Chlebus *et al.*, *Towards Better Understanding of Complex Machine Learning Models Using Explainable Artificial Intelligence (XAI): Case of Credit Scoring Modelling.* University of Warsaw, Faculty of Economic Sciences, 2020.

[108] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[109] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "How can we fool lime and shap? adversarial attacks on post hoc explanation methods," 2019.

[110] L. S. Shapley, "A value for n-person games," *Contributions to the Theory of Games*, vol. 2, no. 28, pp. 307–317, 1953.

[111] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3681–3688.

[112] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim, *The (Un)reliability of Saliency Methods.* Cham: Springer International Publishing, 2019, pp. 267–280. [Online]. Available: https://doi.org/10.1007/978-3-030-28954-6_14

[113] P. Lipton, "Contrastive explanation," *Royal Institute of Philosophy Supplements*, vol. 27, pp. 247–266, 1990.

[114] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[115] F. Wortmann and K. Flüchter, "Internet of things," *Business & Information Systems Engineering*, vol. 57, no. 3, pp. 221–224, 2015.

[116] I. T. Union, "Internet of things global standards initiative," 2012.

[117] F. Firouzi and B. Farahani, *Architecting IoT Cloud.* Cham: Springer International Publishing, 2020, pp. 173–241. [Online]. Available: https://doi.org/10.1007/978-3-030-30367-9_4

[118] J. Diechmann, K. Heineke, T. Reinbacher, and D. Wee, “The internet of things: How to capture the value of iot,” Tech. Rep., 2018.

[119] M. A. J. Jamali, B. Bahrami, A. Heidari, P. Allahverdizadeh, and F. Norouzi, “Iot architecture,” *Towards the Internet of Things*, pp. 9–31, 2020.

[120] M. Burhan, R. A. Rehman, B. Khan, and B.-S. Kim, “Iot elements, layered architectures and security issues: A comprehensive survey,” *Sensors*, vol. 18, no. 9, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/9/2796

[121] M. Mukherjee, I. Adhikary, S. Mondal, A. K. Mondal, M. Pundir, and V. Chowdary, “A vision of iot: applications, challenges, and opportunities with dehradun perspective,” in *Proceeding of International Conference on Intelligent Communication, Control and Devices.* Springer, 2017, pp. 553–559.

[122] Cisco, “Fog computing and the internet of things: extend the cloud to where the things are,” Tech. Rep., 2016.

[123] S. Venticinque and A. Amato, “A methodology for deployment of iot application in fog,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 5, pp. 1955–1976, 2019.

[124] P. Kumar, G. P. Gupta, and R. Tripathi, “A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–18, 2020.

[125] R. Pecori, “A pki-free key agreement protocol for p2p voip applications,” in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 6748–6752.

[126] H. Tahaei, F. Afifi, A. Asemi, F. Zaki, and N. B. Anuar, “The rise of traffic classification in iot networks: A survey,” *Journal of Network and Computer Applications*, vol. 154, p. 102538, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520300126

[127] U. 42, “2020 unit 42 iot threat report,” Palo Alto, Tech. Rep., 2020. [Online]. Available: https://unit42.paloaltonetworks.com/iot-threat-report-2020/

[128] T. Marsden, N. Moustafa, E. Sitnikova, and G. Creech, “Probability risk identification based intrusion detection system for scada systems,” 2017.

[129] V. K. Rahul, R. Vinayakumar, K. Soman, and P. Poornachandran, “Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security,” in *2018 9th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2018.* Institute of Electrical and Electronics Engineers Inc., 10 2018.

[130] R. Vinayakumar, K. P. Soman, and P. Poornachandran, “Applying convolutional neural network for network intrusion detection,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[131] J. Kim, J. Kim, H. L. Thi Thu, and H. Kim, “Long short term memory recurrent neural network classifier for intrusion detection,” in *2016 International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5.

[132] T. Le, J. Kim, and H. Kim, “An effective intrusion detection classifier using long short-term memory with gradient descent optimization,” in *2017 International Conference on Platform Technology and Service (PlatCon)*, 2017, pp. 1–6.

[133] J. Kim and H. Kim, “Applying recurrent neural network to intrusion detection with hessian free optimization,” in *Information Security Applications*, H.-w. Kim and D. Choi, Eds. Cham: Springer International Publishing, 2016, pp. 357–369.

[134] C. Xu, J. Shen, X. Du, and F. Zhang, “An intrusion detection system using a deep neural network with gated recurrent units,” *IEEE Access*, vol. 6, pp. 48 697–48 707, 2018.

[135] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks,” *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017.

[136] A. Andalib and V. T. Vakili, "An autonomous intrusion detection system using an ensemble of advanced learners," in *2020 28th Iranian Conference on Electrical Engineering (ICEE)*, 2020, pp. 1–5.

[137] N. Chaibi, B. Atmani, and M. Mokaddem, "Deep learning approaches to intrusion detection: A new performance of ann and rnn on nsl-kdd," in *Proceedings of the 1st International Conference on Intelligent Systems and Pattern Recognition*, ser. ISPR '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 45–49. [Online]. Available: https://doi.org/10.1145/3432867.3432889

[138] H. Gwon, C. Lee, R. Keum, and H. Choi, "Network intrusion detection based on lstm and feature embedding," *ArXiv*, vol. abs/1911.11552, 2019.

[139] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.

[140] B. Adhi Tama and K. H. Rhee, "Attack classification analysis of iot network via deep learning approach," *Research Briefs on Information Communication Technology Evolution (ReBICTE)*, vol. 3, 11 2017.

[141] S. A. Althubiti, E. M. Jones, and K. Roy, "Lstm for anomaly-based network intrusion detection," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–3.

[142] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18327687

[143] M. A. Ferrag and L. Maglaras, "Deepcoin: A novel deep learning and blockchain-based energy exchange framework for smart grids," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1285–1297, 2020.

[144] S. Aldhaheri, D. Alghazzawi, L. Cheng, B. Alzahrani, and A. Al-Barakati, “Deep-dca: novel network-based detection of iot attacks using artificial immune system,” *Applied Sciences*, vol. 10, no. 6, p. 1909, 2020.

[145] Y. N. Soe, P. I. Santosa, and R. Hartanto, “Ddos attack detection based on simple ann with smote for iot environment,” in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 2019, pp. 1–5.

[146] A. Derhab, A. Aldweesh, A. Z. Emam, and F. A. Khan, “Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering,” *Wireless Communications and Mobile Computing*, vol. 2020, 2020.

[147] M. Ge, X. Fu, N. Syed, Z. Baig, G. Teo, and A. Robles-Kelly, “Deep learning-based intrusion detection for iot networks,” in *2019 IEEE 24th Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2019, pp. 256–25 609.

[148] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, “Ton_iot: The role of heterogeneity and the need for standardization of features and attack types in iot network intrusion datasets,” *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[149] N. Moustafa, “A new distributed architecture for evaluating ai-based security systems at the edge: Network ton_iot datasets,” *Sustainable Cities and Society*, vol. 72, p. 102994, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210670721002808

[150] S. R. Islam, W. Eberle, S. K. Ghafoor, A. Siraj, and M. Rogers, “Domain Knowledge Aided Explainable Artificial Intelligence for Intrusion Detection and Response,” *arXiv*, 11 2019. [Online]. Available: http://arxiv.org/abs/1911.09853

[151] D. L. Marino, C. S. Wickramasinghe, and M. Manic, “An adversarial approach for explainable ai in intrusion detection systems,” in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 2018, pp. 3237–3243.

[152] H. Li, F. Wei, and H. Hu, “Enabling dynamic network access control with anomaly-based ids and sdn,” in *Proceedings of the ACM International Workshop on Security in Software Defined Networks Network Function Virtualization*, ser. SDN-NFVSec ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 13–16. [Online]. Available: https://doi.org/10.1145/3309194.3309199

[153] W. Guo, D. Mu, J. Xu, P. Su, G. Wang, and X. Xing, “Lemna: Explaining deep learning based security applications,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 364–379. [Online]. Available: https://doi.org/10.1145/3243734.3243792

[154] L. Xiao, X. Wan, X. Lu, Y. Zhang, and D. Wu, “Iot security techniques based on machine learning: How do iot devices use ai to enhance security?” *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 41–49, 2018.

[155] I. Kim, S. Rajaraman, and S. Antani, “Visual interpretation of convolutional neural network predictions in classifying medical image modalities,” *Diagnostics*, vol. 9, no. 2, p. 38, 2019.

[156] S. Shi, X. Zhang, and W. Fan, “Explaining the predictions of any image classifier via decision trees,” *arXiv preprint arXiv:1911.01058*, 2019.

[157] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud, “Explaining image classifiers by counterfactual generation,” *arXiv preprint arXiv:1807.08024*, 2018.

[158] S. M. Mathews, “Explainable Artificial Intelligence Applications in NLP, Biomedical, and Malware Classification: A Literature Review,” in *Advances in Intelligent Systems and Computing*, vol. 998. Springer Verlag, 7 2019, pp. 1269–1292. [Online]. Available: https://doi.org/10.1007/978-3-030-22868-2_90

[159] J. Srinivas, A. K. Das, and N. Kumar, “Government regulations in cyber security: Framework, standards and recommendations,” *Future Generation Computer Systems*, vol. 92, pp. 178–188, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18316753

[160] N. Moustafa, G. Misra, and J. Slay, “Generalized outlier gaussian mixture technique based on automated association features for simulating and detecting web application attacks,” *IEEE Transactions on Sustainable Computing*, 2018.

[161] H. Hindy, R. Atkinson, C. Tachtatzis, J.-N. Colin, E. Bayne, and X. Bellekens, “Utilising deep learning techniques for effective zero-day attack detection,” *Electronics*, vol. 9, no. 10, p. 1684, 2020.

[162] C. Wu, A. Qian, X. Dong, and Y. Zhang, “Feature-oriented design of visual analytics system for interpretable deep learning based intrusion detection,” in *2020 International Symposium on Theoretical Aspects of Software Engineering (TASE).* IEEE, 2020, pp. 73–80.

[163] A. Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, “A survey of neural networks usage for intrusion detection systems,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 497–514, 2021.

[164] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, “Deep recurrent neural network for iot intrusion detection system,” *Simulation Modelling Practice and Theory*, vol. 101, p. 102031, 2020.

[165] B. Zhou, Y. Sun, D. Bau, and A. Torralba, “Interpretable basis decomposition for visual explanation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[166] S. Anjomshoae, K. Främling, and A. Najjar, “Explanations of black-box model predictions by contextual importance and utility,” in *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems.* Springer, 2019, pp. 95–109.

[167] N. Seedat, V. Aharonson, and Y. Hamzany, “Automated and interpretable m-health discrimination of vocal cord pathology enabled by machine learning,” in *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE).* IEEE, 2020, pp. 1–6.

[168] T. R. Wood, C. Kelly, M. Roberts, and B. Walsh, “An interpretable machine learning model of biological age,” *F1000Research*, vol. 8, no. 17, p. 17, 2019.

[169] R. V. Kumar Reddy, B. Srinivasa Rao, and K. P. Raju, “Handwritten hindi digits recognition using convolutional neural network with rmsprop optimization,” in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2018, pp. 45–51.

[170] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[171] S. Bock and M. Weiß, “A proof of local convergence for the adam optimizer,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[172] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy.” New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2976749.2978318

[173] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[174] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the kdd cup 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[175] L. Dhanabal and S. Shantharajah, “A study on nsl-kdd dataset for intrusion detection system based on classification algorithms,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[176] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, “Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems,” *IEEE Access*, vol. 8, pp. 165 130–165 150, 2020.

[177] N. Pham, N. Moustafa, and B. Turnbull, "Github - buncha-cob/xai-ids: Explainable ai-based ids in iot networks," 2021. [Online]. Available: https://github.com/bunCha-cob/XAI-IDS

[178] W. Alhakami, A. ALharbi, S. Bourouis, R. Alroobaea, and N. Bouguila, "Network anomaly intrusion detection using a nonparametric bayesian approach and feature selection," *IEEE Access*, vol. 7, pp. 52 181–52 190, 2019.

[179] S. Smith, "Iot connections to reach 83 billion by 2024, driven by maturing industrial use cases," 2021.

[180] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on big data environment," *Journal of Big Data*, vol. 5, no. 1, pp. 1–12, 2018.

[181] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Computer networks*, vol. 174, p. 107247, 2020.

[182] I. Channoufi, S. Bourouis, N. Bouguila, and K. Hamrouni, "Color image segmentation with bounded generalized gaussian mixture model and feature selection," in *2018 4th International conference on advanced technologies for signal and image processing (ATSIP)*. IEEE, 2018, pp. 1–6.

[183] ——, "Spatially constrained mixture model with feature selection for image and video segmentation," in *International Conference on Image and Signal Processing*. Springer, 2018, pp. 36–44.

[184] S. Bourouis, Y. Laalaoui, and N. Bouguila, "Bayesian frameworks for traffic scenes monitoring via view-based 3d cars models recognition," *Multimedia Tools and Applications*, vol. 78, no. 13, pp. 18 813–18 833, 2019.

[185] I. S. Thaseen, C. A. Kumar, and A. Ahmad, "Integrated intrusion detection model using chi-square feature selection and ensemble of classifiers," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3357–3368, 2019.

[186] W. Zong, Y.-W. Chow, and W. Susilo, “A two-stage classifier approach for network intrusion detection,” in *International Conference on Information Security Practice and Experience.* Springer, 2018, pp. 329–340.

[187] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, “Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model,” *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[188] T. Shawly, M. Khayat, A. Elghariani, and A. Ghafoor, “Evaluation of hmm-based network intrusion detection system for multiple multi-stage attacks,” *IEEE Network*, vol. 34, no. 3, pp. 240–248, 2020.

[189] C. Molnar, G. Casalicchio, and B. Bischl, “Interpretable machine learning–a brief history, state-of-the-art and challenges,” *arXiv preprint arXiv:2010.09337*, 2020.

[190] M. L. Head, L. Holman, R. Lanfear, A. T. Kahn, and M. D. Jennions, “The extent and consequences of p-hacking in science,” *PLOS Biology*, vol. 13, no. 3, pp. 1–15, 03 2015. [Online]. Available: https://doi.org/10.1371/journal.pbio.1002106