Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Exploring Explainable Artificial Intelligence (XAI) in Telecommunication Networks

Author:
Pieter BARNARD

Supervisor:
Prof. Nicola MARCHETTI

Co-Supervisors:
Prof. Luiz A. DASILVA
Dr. Irene MACALUSO

This thesis is submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the Department of Electronic & Electrical Engineering

March 2025

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish.

Elements of this work that have been carried out jointly with others or by collaborators have been duly acknowledged in the text wherever included.

Date

Pieter Barnard

Abstract

Artificial intelligence (AI) is poised to become a key technology in future telecommunication networks due to its profound ability to solve complex tasks in a data-driven and, in some cases, model-free manner. Although AI has proven its ability to yield superior performance over traditional optimisation approaches in many other engineering disciplines, such as audio, video and image processing, etc., its adoption into key critical areas of the telecommunication domain remains severely hampered. One of the main barriers to the widespread adoption of AI in the telecommunication domain stems from an inherent lack of understanding of how complex AI models formulate their decisions and the potential implications this has towards the safe and reliable operation of the network. Recently, the field of eXplainable AI (XAI) has emerged as a promising area of research that can help overcome the "black-box" limitation associated with complex AI models by providing practitioners and other stakeholders with explanations as to how the AI model arrives at its final decisions.

In this thesis, we explore some of the potential benefits that XAI can have within the telecommunication domain. Specifically, we focus on two key areas of the networking ecosystem, where the general complexity of the system motivates both the adoption of AI-driven solutions and where the critical nature of the problem means that transparency becomes a crucial requirement to ensure any decisions of the proposed solution can be thoroughly validated and trusted by its affected stakeholders. The main research question we address in this thesis can be summarized as follows: "How can explainable artificial intelligence (XAI) be integrated into high-stake areas of the telecommunication domain to ensure better reliability and robust service within the network?". In addressing this research question, the first part of this thesis explores the problem of network slicing, where a network operator may lease specific slices of its network resources and infrastructure to secondary tenants. Within this particular research domain, we first propose and compare various AI-based solutions for optimising the network resources scheduled

to the slice tenant. We then augment our proposed framework with additional layers of explainability and demonstrate how such explanations can be used to gain important insights into the model's decisions, as well as to aid the human-in-the-loop to debug and devise targeted solutions to fix any potential misbehaviour of the model.

In the second part of this thesis, we explore some of the benefits that XAI can bring within the cybersecurity space. We begin by proposing a novel methodology for explainable intrusion detection. In contrast to existing works on this topic, our proposed methodology includes an additional layer of unsupervised machine learning (ML), which takes the explanations from an initial network intrusion detection system (NIDS) as input and leverages the information gained from these explanations to automatically enhance the performance of the overall intrusion system against zero-day attacks. Finally, we extend our methodology to support general improvements in the overall accuracy of the system against any attack type and perform an in-depth analysis to evaluate how well our methodology generalizes to cases where i) multiple intrusion datasets are considered, ii) the choice of model used by the initial NIDS is varied, iii) the choice of XAI methodology is varied, and iv) we assess the ability of our methodology to generalise to unseen data distributions.

Acknowledgements

I would like to dedicate this short chapter to all those who have helped and guided me during my PhD journey. To me, this journey has been quite a long one filled with many ups and downs and the occasional twist. Throughout this journey, a number of people have played a pivotal role in supporting me and convincing me to never give up.

Before all, I would like to thank my parents for all their love and support, and for always pushing me to do my best. I also want to thank my brother for all the useful advice he has given me over the years. I would also like to give a special thanks to all of my loving grandparents, who, since my earliest memories, all believed that I would one day be successful.

First and foremost, I would like to express my sincere gratitude to my supervisors, who have guided me so much over the past few years. In chronological order, I would like to thank Prof. Luiz DaSilva, who served as my main PhD supervisor during the first few months of my PhD, as well as having previously served as my supervisor during my bachelor's degree; I've learnt so much from all your expert advice over the years and hope to one day pass on similar advice to others during their own journeys. Secondly, I would like to express my sincere gratitude to my main supervisor, Prof. Nicola Marchetti, who agreed to take over as my main supervisor when Luiz returned to the US; although almost all of our meetings ended up being online, you still managed to instil a great deal of knowledge and expertise into all of my research and always provided useful comments and suggestions to help me along the way. Lastly, I want to thank Dr. Irene Macaluso, who although only served for a brief period as one of my co-supervisors, still provided me with some of the most useful advice during my PhD journey, and ultimately helped me to finalise my first paper as a primary author; I wish you all the best in your current career and hope to one day thank you in person.

I would like to thank all of my friends and colleagues over in the CONNECT centre and the Electrical & Electronic Department of Trinity. In particular, Jean-Baptiste, Jernej,

Boris, Erika, Fadhil, Abel, Cong, Vivek, Arijeet, Yan, Bruno, Danh, Linh and many more. Finally, a special thanks to my loving girlfriend, Maryam, who has helped and supported me so much over the last few months of my PhD.

Contents

List of Acronyms

1G	First Generation
2G	Second Generation
3G	Third Generation
4G	Fourth Generation
5G	Fifth Generation
B5G	Beyond 5G
AFA	Additive Feature Attribution
AI	Artificial Intelligence
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
DNN	Deep Neural Network
IoT	Internet of Things
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
XRL	Explainable Reinforcement Learning
E2E	End-to-End
eMBB	enhanced Mobile Broadband
EU	European Union
GRU	Gated Recurrent Unit
KPI	Key Performance Indicator
LSTM	Long Short-Term Memory
RRM	Radio Resource Management

ML Machine Learning

mMTC massive Machine-Type Communications

MNO Mobile Network Operator

MSE Mean-Square Error

NFV Network Function Virtualisation

VAR Virtual and Augmented Reality

OTT Over-The-Top

PCA Principle Component Analysis

QoE Quality of Experience

QoS Quality of Service

QoT Quality of Trust

RMSE Root Mean-Square Error

RSRP Reference Signal Received Power

SDN Software-Defined Networking

SGD Stochastic Gradient Descent

STRR Short-Term Resource Reservation

SLA Service-Level Agreement

SP Service Provider

SVM Support-Vector Machine

uRLLC ultra-Reliable and Low-Latency Communications

XAI Explainable AI

NIDS Network Intrusion Detection System

MIMO Multiple Input Multiple Output

ODFM Orthogonal Frequency-Division Multiplexing

NOMA Non-Orthogonal Multiple Access

SMS Short Message Service

VoIP Voice over Internet Protocol

STEM Science, Technology, Engineering, and Mathematics

NPC non-player character

ICT Information and Communications Technology

SDO Standards Development Organization

ZSM Zero-Touch Network and Service Management

ENI Experiential Network Intelligence

ETSI European Telecommunications Standards Institute

ITU-T International Telecommunication Union Telecommunication Standardization Sector

HITL human-in-the-loop

ARIMA Autoregressive Integrated Moving Average

MDP Markov decision process

BER Bit Error Rate

XGBoost Extreme Gradient Boosting

DAE Deep Auto-encoder

PoV Percentage of Variance

SHAP SHapley Additive exPlanation

LIME Local Interpretable Model-agnostic Explanation

LRP Layer-wise Relevance Propagation

ARE Absolute Reconstruction Error

MPGNN Message-Passing Graph Neural Network

WMMSE Weighted Minimum Mean-Square Error

Grad-CAM Gradient-weighted Class Activation Mapping

I/Q in-phase and quadrature

WiFi Wireless Fidelity

MCS Modulation and Coding Scheme

CSI Channel State Information

RRT Round-Trip Time

ReLu Rectified Linear Unit

MMSE Minimum Mean-Squared Error

QPSK Quadrature Phase-Shift Keying

WBFM Wide Band Frequency Modulation

QAM Quadrature Amplitude Modulation

GFSK Gaussian Frequency Shift Keying

AP Access Point

LLM Linear Mixed Models

AUC Area Under the Curve

MLP Multilayer Perceptron

RM Resource Management

UAV Unmanned Aerial Vehicle

MAPLE Model Agnostic Supervised Local Explanation

Chapter 1

Introduction

In this chapter, we introduce and discuss the main research topics explored in our work and how they relate to the overall contributions of this thesis. Specifically, we begin by outlining the main motivation of our research. We then describe the overall aims and objectives of our work and state the research questions and contributions that this thesis aims to address. Finally, we list each of the publications associated with this thesis and briefly outline how each publication pertains to the contributions and research questions addressed in this thesis.

1.1 Motivation

Since the beginning of the digital era, telecommunication networks have been constantly evolving and adapting to meet the growing demands of our society. This never-ending transformation can be best exemplified through the numerous generations of telecommunication networks that have emerged over the past five decades, with each new generation offering substantially improved Key Performance Indicators (KPIs) and functionality than the last. For example, the First Generation (1G) of telecom networks, which dominated the telecom landscape throughout the late 1970s and 1980s, provided analogue voice calls to its end users. Roughly one decade later, the emergence of the Second Generation (2G) brought about the first transition to digital voice call services in addition to basic Short Message Service (SMS) and low-speed data service. By the early 2000s, the Third Generation (3G) of telecom networks expanded upon the functionality of 2G by offering even higher data speeds as well as enhanced voice services to support emerging use cases, such as video calling and multimedia applications. Perhaps as a revolutionary turning point, the Fourth Generation (4G), which emerged roughly one decade after

3G's first appearance, saw many new and exciting technologies, such as Multiple Input Multiple Output (MIMO), Orthogonal Frequency-Division Multiplexing (ODFM), small-cells (e.g., femtocells and picocells), etc., being integrated into the network infrastructure. These new technologies undoubtedly lent a helping hand towards the proliferation and widespread success of emerging smartphones during this period, which made use of 4G capabilities to provide high-speed streaming services and applications to its end users, such as online gaming, high-quality video calls, Voice over Internet Protocol (VoIP), etc. Finally, the most recent generation, i.e., Fifth Generation (5G), which became commercially available in 2019 and continues to be rolled out across various parts of the globe, has built upon the success of 4G by introducing even more key technologies into the network ecosystem, such as massive MIMO, network slicing, millimetre-wave communications [1], dense small-cells, edge computing etc. Together, these technologies allow 5G to offer even lower latency, higher capacity and greater reliability compared to 4G, while being able to support a wide host of novel use cases and services centred around massive Machine-Type Communications (mMTC) (i.e., such as large networks of Internet of Things (IoT) devices, smart wearables, etc.), enhanced Mobile Broadband (eMBB) (i.e., used by high bandwidth applications, such as ultra-high-definition video streaming, Virtual and Augmented Reality (VAR), etc.), and ultra-Reliable and Low-Latency Communications (uRLLC) (i.e., used by critical services, such as remote surgery, autonomous vehicles, etc.)

As the development of 5G technologies continues to evolve across many parts of the globe, a new arms race is taking shape as researchers from both sides of the academic and industrial fence look towards the next generation of telecommunication networks, which are expected to comprise an even larger consortium of novel use cases with substantially more diverse and stringent KPIs [2]. For example, some of these new and emerging services are expected to stem from business models and vertical markets, such as e-health, smart cities, smart factories, autonomous vehicles, etc. all of whom will require exponential improvements in terms of network KPIs, such latency, capacity, reliability, energy efficiency, security and privacy [3]–[5].

As one of the key enablers to address the ambitious targets set by future networks, Artificial Intelligence (AI) is poised to revolutionise the way in which future networks will be monitored, optimised and proactively managed [6]–[8]. Besides, AI and Machine

Learning (ML) based algorithms have already shown tremendous success in advancing the state-of-the-art in many other related fields of Science, Technology, Engineering, and Mathematics (STEM) research. For instance, in the biological and medical disciplines, AI-driven algorithms have been used to discover new types of medicines [9], to accurately predict disease outcomes [10], as well as to enhance our overall understanding of complex biological and genetic processes [11]. Similarly, in fields of engineering and computer science, AI and ML-based algorithms have been at the forefront of multiple breakthroughs in recent years, in some cases, even demonstrating superhuman performance across a wide range of tasks, including image recognition [12], [13], content understanding [14], language translation, and sentiment analysis, strategic games [15], etc. Meanwhile, in the telecom domain, the adoption of AI is still within its infancy stage, with the majority of real-world use cases being limited to non-critical applications, such as AI-driven customer service chat-bots, personalised recommendation systems, network planning guidance, spam filtering [16]–[18], etc., while the adoption of AI into potentially more impactful - but also more critical – areas of the networking domain, such as autonomous network and resource management, vehicle-to-vehicle communications [19], operations, administration and management (OAM) [20], etc., has remained severely hampered in real-world deployments, despite the significant performance gains that AI solutions may offer over existing conventional optimisation approaches (e.g., rule-based policies, static threshold policies, heuristic algorithms, mathematical and model-dependent techniques etc.). To compound this issue further, many researchers and industries in the Information and Communications Technology (ICT) sector agree that the complexity of the current 5G ecosystem is already at the tipping point of what operators can successfully manage and maintain, while the data-driven and model-free nature associated with AI algorithms may offer a potential solution to help overcome the impending complexity crisis faced by future networks [7], [21].

One of the main contributors to the reluctant uptake of AI/ML into critical components of the telecommunications domain stems from the inability of operators and other stakeholders of the network to understand the reasoning processes and decision-making policies of complex AI/ML algorithms. This effect has led to many complex AI/ML models, such as Deep Neural Networks (DNNs) and ensemble tree-based models, etc., being infamously regarded as "black-boxes". Especially in critical areas of the telecom

domain, the inability of operators to adequately understand and validate the actions of their AI/ML models poses a serious threat to the reliable and trustworthy operation of the network. Thus, if future networks truly intend to reap the benefits that AI can offer to the network ecosystem, such as autonomous network management, enhanced security, improved Quality of Service (QoS), etc., then it becomes crucially important to first develop suitable methodologies that can help overcome the limitations posed by complex black-box algorithms, and thereby, foster greater transparency, trust, and reliability into the network.

In this thesis, we explore one potential solution that can help alleviate the challenges posed by black-box AI/ML algorithms deployed within the telecom domain. Namely, we turn to the field of Explainable AI (XAI), which is a relatively new and emerging branch of AI that aims to bring transparency to the use of black-box models by providing stakeholders with the ability to examine the inner mechanisms of the model and extract useful explanations describing *why* and *how* the model makes its decisions [22]–[24]. In the next subsection, we provide a brief overview of some of the key concepts associated with XAI as well as discuss the role that this emerging technology is expected to fulfil in future networks. This brief background serves as the foundation to understand and appreciate the motivation and relevance behind the proposed research questions addressed in this thesis.

1.2 Explainable Artificial Intelligence (XAI)

1.2.1 Brief History & Definition

As early as the late 1970s and early 1980s, some researchers have been working on the idea of enriching expert-based systems with the capability of explaining their own actions and behaviour based on the set of rules they discover during their training. Such research was mostly confined to applications where there was a definitive need for humans to understand how their models derived their actions and recommendations based on the specific input conditions presented to the model, such as in medical diagnosis [25], educational programming [26], and logic debugging [27]. However, it wasn't until later in the early 2000s when the first use of the phrase "XAI" was originally coined by

the authors in [28], who made reference to this term when describing the ability of AI-driven non-player characters (NPCs) to explain their actions within a simulation game for military training [22]. Since this time, the concept of explainability has witnessed a remarkable resurgence of interest from the academic and industry research communities, as well as from various policymakers around the world. As a simple indication of this renewed interest, Figure 1.1 shows a trend line of the number of searches for the term "explainable AI" across the Google search engine over the last two decades. From this figure, it is clear to see that the concept of XAI has since expanded to become a dominant field of research, while it continues to grow at an exceptionally fast pace.

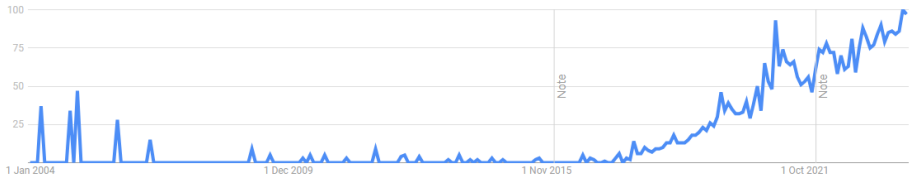


FIGURE 1.1: Trend line depicting the number of searches for the term "explainable AI" on Google's search engine for the past two decades.

The works of [22], [29]–[31] provide a general definition for the term "explainable AI", which we also adopt throughout the scope of this thesis. In these works, "eXplainable AI" or XAI is defined as a field of research that i) aims to develop a suite of techniques and methodologies that can be used to examine and expose the underlying behaviour of black-box models, where ii) this behaviour is reflected in the form of human-understandable explanations, such that iii) the process of generating explanations does not inadvertently alter the performance of the model. While this definition captures the general idea about what XAI entails, a number of additional classifications have been proposed based on the scope and assumptions made by a particular XAI methodology. In Chapter 2, we introduce some of the main classifications and finer distinctions that form part of the general XAI taxonomy used throughout the literature, as well as provide a brief discussion on how XAI relates to similar topics in AI, such as trust, fairness, and accountability. In the next subsection, we briefly discuss some of the regulatory implications of XAI within the telecommunication industry.

1.2.2 The role of XAI in AI Regulations

As our society continues to become more reliant on AI technologies across various domains, such as healthcare, finance, retail, manufacturing and the ICT sector, etc. so does the need for ensuring that greater responsibility and transparency in AI practices are enforced within these critical domains. Recently, the on-going proliferation of AI-driven technologies into almost all sectors of our society, coupled with the potential risks that these technologies pose to human and societal rights, have spurred numerous policymakers around the globe to develop new regulations specifically targeting the deployment of AI technologies. For example, article 22 of the General Data Protection Regulation (GDPR), which came into effect in 2018 and governs the data protection laws across the European Union (EU), empowers individuals with the right to request an explanation, express their point of view, and contest any automated decisions based on their data. In the context of telecommunications, this could include decisions related to credit scoring, service delivery, or even targeted advertising. In this sense, explanations based on feature importance or "what-if" analysis could provide users with understandable insights into how these decisions are made, facilitating their right to understand and potentially challenge possible biases. In addition, the recently proposed AI Act [32], which is currently in draft form at the time of writing, seeks to ensure that any AI technologies developed and deployed within EU member states are free of unwanted biases, provide transparency in their decisions and the data used to train them, as well as foster trust, responsibility, and accountability through imposing strict requirements, especially within "high-risk" applications. Moreover, as an important aspect, the AI act explicitly acknowledges the importance of XAI as a potential tool that can enhance the transparency and trustworthiness of AI technologies based on the interpretable explanations it can provide to stakeholders regarding the decision-making processes of these technologies.

In the telecommunications sector, Standards Development Organizations (SDOs), such as the European Telecommunications Standards Institute (ETSI) and International Telecommunication Union Telecommunication Standardization Sector (ITU-T), have also recently started to progress towards addressing the challenges of integrating AI into critical areas of future networks. On this front, while there still doesn't appear to be a clear-cut or

definitive answer in terms of the specific role and scope that XAI will play in future AI-driven networks, it is evident that traction on this topic is beginning to gather pace within the telecommunications and ICT sector. For example, ETSI's Experiential Network Intelligence (ENI) working group [33], [34], which focuses on the use of AI and ML for cognitive network management, touches on multiple topics directly related to the concept of XAI, including for instance, the use of knowledge representation via data analysis to train and explain AI models, the proposal of cognitive models that can explain their behaviour and learn from external feedback, and the need for developing AI models that can understand cause-and-effect relationships to allow the model to be easily explained to humans. As another example, the ETSI working group for Zero-Touch Network and Service Management (ZSM), in response to the EU's "Proposal for a Regulation laying down harmonised rules on artificial intelligence" [35], have explicitly incorporated ideas of explainability into the policies and definitions of their specification. This includes, for instance, the adoption of various definitions for interpretable ML, fair ML, trustworthy ML (including the proposal of a Quality of Trust (QoT) metric to measure the trustworthiness of an AI system), a categorisation of three risk levels for a given application using AI/ML (e.g., limited risk, high-risk, and unacceptable risk), and a further seven key requirements for a ML system to be considered trustworthy (e.g., human oversight, robustness, privacy, transparency, fairness, accountability, societal well-being). Moreover, this initiative also proposes the idea of "AI enablers" as ML models which can support automated network management while providing explanations for their actions.

The ITU-T has also recently begun highlighting the need for greater explainability in AI-driven systems, especially within the scope of cybersecurity. For example, the ITU-T technical report on the "Guidelines for security management of using artificial intelligence technology" explicitly mentions the need for explaining the behaviour of AI systems as one of six top security objectives of organisations that adopt AI. Moreover, the report emphasises the importance of ensuring traceability of the decisions made by AI systems so that the cause of security incidents can be more accurately found. In this sense, explainable AI could be leveraged alongside methods of causal inference to offer greater traceability support and to help improve security analysis and incident response.

1.3 Thesis Aims and Objectives

In this thesis, we explore some of the challenges and benefits that explainability can offer to the development of complex AI/ML models used in the telecommunication domain. Specifically, the first and primary research question that this thesis aims to address can be summarised as follows:

RQ1: How can explainable artificial intelligence (XAI) be integrated into high-stake areas of the telecommunication domain to ensure better reliability and robust service within the network?

In addressing this question, we first identify two key areas of the telecom domain where AI/ML is actively being pursued by the academic and industry research communities, as well as where the high-stakes nature of each setting provides a sound motivation for the need for understanding how complex solutions being developed in each setting behave. Namely, the first setting that we explore relates to the problem of resource management within the context of network slicing. Within this problem area, we first develop and assess a suite of AI models for resource allocation. We then develop an extended framework for providing explanations of various types to demystify and explain the behaviour of an ML model for resource reservation. Essentially, our framework allows operators to understand the behaviour of the model at both a local level, i.e., understand the factors behind a single prediction of the model, as well as at a global level, i.e., understand how different factors affect the model's behaviour across a broader context. Moreover, by manually analysing the information conveyed in these types of explanations, we demonstrate how operators can debug and rectify potentially unwanted behaviours of the reservation model. As a consequence of this, operators can improve the overall performance and reliability of their final model before it is deployed into a real-world environment. Finally, since the benefits we can expect to reap from our methodology depend on the ability of the underlying explanations to faithfully capture and describe the model's true behaviour, we also develop and present a suite of techniques to quantitatively assess the faithfulness of the explanations produced by our framework. By enabling operators and other stakeholders of the network to assess and validate the faithfulness of the explanations they are presented with, this final step ensures that our methodology remains actionable and trustworthy in a real-world context.

We summarise the relevance of this final step in our second research question:

RQ2: How can we quantitatively evaluate the faithfulness of the explanations to ensure that our proposed framework remains trustworthy and actionable to world environments?

The second area that we explore relates to the field of cybersecurity, and in particular, network intrusion detection. Within this domain, our first objective is to augment the overall capabilities of our previously proposed framework by addressing the following, and third research question:

RQ3: How can we minimise the effort of the human-in-the-loop (HITL) by automatically analysing the information contained within the explanations and use this to improve the system's performance?

Several reasons motivate our curiosity to answer this question. First, we envision that future networks will become increasingly managed by automated decisions, which are typically accomplished through the use of AI/ML models. In the cybersecurity domain, in particular, an already vast and continuously expanding body of research is appearing on the use of AI for security-related topics, such as network intrusion detection, malware detection, binary reverse engineering, etc. In this sense, complex AI solutions are being proposed to replace conventional cybersecurity approaches, such as hand-crafted rules or simplistic models (i.e., shallow decision trees, linear models, etc.), which tend to lack adequate performance and/or require constant retraining to keep up with the ongoing efforts of attackers, who continually strive to develop new ways to exploit the vulnerabilities of the network. Given the widespread adoption of AI into this domain, and coupled with growing legal pressures received from governmental agencies calling for more transparency in the use of AI models, we chose this problem area as the use case for our third research question.

Another factor that motivates our choice stems from the fact that, in a network intrusion setting, vast quantities of data packets typically pass through even a single node of the network, and within a short duration of time. Thus, a correspondingly large number of outputs (as well as their corresponding explanations) are expected to be generated by a Network Intrusion Detection System (NIDS). In this sense, we strongly believe it would become unfeasible for any human-in-the-loop (HITL) to analyse and make sense of each

explanation produced by the model in the first place, which further motivates our third research question and chosen use case scenario.

Lastly, we argue that if the information conveyed in each explanation is meant to inform a human counterpart about some interesting aspect of the model's behaviour, then this same information should also be useful to additional ML approaches, which are already renowned for their abilities in extracting and analysing large quantities of information in real-time. To elaborate this point further, Figure 1.2 shows a typical pipeline from the literature where a HITL is incorporated to analyse and process explanations of an ML system. In this sense, we envision that the human themselves acts as a complex model in this system, who uses their own form of logic and reasoning to make sense of the explanation and to decide if it is necessary to take any action. From this perspective, we propose to augment the reasoning machine of the human (i.e., the brain) by mimicking, to a limited extent, its role in the explanation task using a ML model, such as a DNN, which can be easily trained to detect explanations of interest and improve the system's performance.

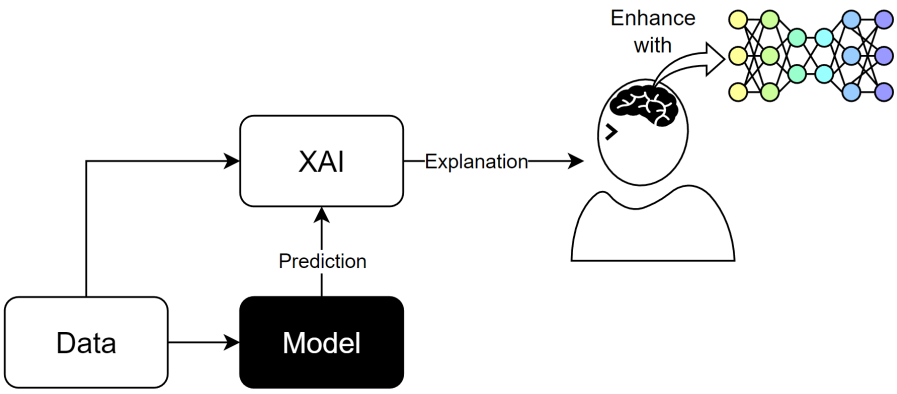


FIGURE 1.2: Simplified overview of our proposal to address RQ3.

In this context, we first propose a methodology for network intrusion detection in which we further propose the use of an unsupervised learning approach, i.e., such as a deep autoencoder model, to replace the HITL in the conventional ML pipeline for intrusion detection. As part of our overall solution, we demonstrate how the use of additional ML to analyse the explanations can be leveraged to accurately detect zero-day attacks (i.e., attacks that the system has never seen during its training) as well as significantly augment the performance capabilities of our overall NIDS.

While these initial results provide a promising indication of our methodology's capabilities, we next extend the scope of our methodology to support general improvement gains in the overall performance of the system and devise a series of in-depth experiments to evaluate how well our methodology generalizes to cases where i) multiple intrusion datasets are considered, ii) the choice of model used by the initial NIDS is varied, iii) the choice of XAI methodology is varied, and iv) we assess the ability of our methodology to generalise to unseen data distributions. Finally, we summarise the outcome of this work in our fourth research question:

RQ4: How can we verify the performance gains of our proposed methodology used to address RQ3?

1.4 Thesis Contribution

The main contribution of this thesis is the proposal of a generalised methodology that uses XAI and additional ML to automatically improve the performance of complex algorithms deployed in high-stakes telecommunication settings, such as cybersecurity. During this process, we develop various ML/AI and XAI solutions that demonstrate and evaluate our ideas. The source codes for each of these solutions are made publically available and can be found online at our GitHub repository[1]. The overall findings of this thesis support our main claim, which states that while XAI can serve as a useful tool that provides the HITL with valuable insights into the behaviour of the AI/ML algorithm, it can also be employed alongside additional ML to significantly improve the performance of the AI/ML system in applications, such as network intrusion detection. Our main contributions include:

- C1: this contribution addresses directly RQ2 and partially addresses RQ1. First, we develop a suite of AI/ML algorithms for resource reservation within a sliced network environment. Namely, we develop two solutions based on Long Short-Term Memory (LSTM) and vanilla DNN models and compare the performance of each solution against a baseline approach which uses the classical Autoregressive Integrated Moving Average (ARIMA) model. We then develop a framework for

[1] `https://github.com/barnardp`

providing explainability to a general ML/AI model for resource reservation. Our proposed framework includes two types of explanations capturing behaviour at both a local and global level, as well as an evaluation stage, which quantitatively assesses the faithfulness of the explanations against a ground truth.

- C2: this contribution directly addresses RQ3 and partially addresses RQ1. Here, we propose as a first, the idea of passing explanations of an intrusion detection model into an additional AI/ML model for further processing. Specifically, our proposed approach makes use of an unsupervised ML solution, in this case using a deep autoencoder, to automatically learn to distinguish between known and unknown behaviours of the intrusion model. Moreover, we demonstrate using the popular NSL-KDD network intrusion dataset, that the additional ML solution is able to accurately detect zero-day attacks that the intrusion model is unable to detect. By combining the outputs from both the intrusion model and additional autoencoder, we are able to significantly enhance the performance of the overall NIDS.

- C3: this contribution address RQ4 and RQ1. Here, we extend the scope of our proposed methodology from C2, such that the additional autoencoder stage can be used to detect both zero-day attacks as well as attacks of known class but which the intrusion model may fail to detect. In particular, we devise a series of experiments that collectively allows us to assess and validate the generalisation capability of our final proposed methodology under various cases, including where i) multiple intrusion datasets are considered, ii) the type of AI/ML model used by the intrusion model is varied, iii) the choice of XAI methodology is varied, and iv) we assess the ability of our methodology to generalise to unseen data distributions.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 - Review of Related Work & Background Theory - This chapter begins by reviewing main background theory relevant to the work explored in this thesis, including namely a discussion on ML/AI concepts, and concepts related to XAI. Following the background section of this chapter, the next section reviews any related work, including

a broad overview of the telecommunications landscape where concepts of explainability and interpretability have been explored by the research community, followed by a more focused overview on the use of AI/ML and XAI within the areas of resource management and network intrusion detection.

Chapter 3 - Resource Reservation within Sliced Networks - In this chapter we introduce our first contribution, C1, which directly addresses RQ2 and partially addresses RQ1. Specifically, we begin this chapter by proposing two deep-learning solutions for resource allocation within a sliced network environment. We then extend our initial work by proposing a framework leveraging state-of-the-art XAI methodologies to explain and enhance the transparency of black-box models employed in resource management tasks for sliced networks, such as resource reservation. As part of our solution, we demonstrate how explanations from our framework can allow operators to monitor the real-time decisions of their model, understand the global behaviour of their models, as well diagnose potential flaws in the model's decision-making process. Finally, we present an evaluation strategy to assess the faithfulness of the explanations to ensure they remain actionable in a real-world context.

Chapter 4 - Robust Network Intrusion Detection through XAI - This chapter presents our second contribution, C2, which directly addresses RQ3 and partially addresses RQ1. In this chapter, we expand parts of our previous framework from Chapter 3 into the cybersecurity space and propose a novel methodology that leverages XAI alongside additional ML to enhance the overall performance of a NIDS. Our overall results show that our methodology is capable of significantly enhancing the ability of our system to detect zero-day attacks, while its overall performance is comparable to numerous state-of-the-art works from the cybersecurity literature.

Chapter 5 - A General Methodology for Improving Network Intrusion Detection through XAI - This chapter presents our third and final contribution, C3, which addresses RQ4 and RQ1. In this chapter, we first extend our previous methodology from Chapter 4 to support general improvements in the overall accuracy of the intrusion detection system against any attack type. We then perform an in-depth analysis to evaluate how well our methodology generalises under various conditions, including i) when multiple intrusion datasets are considered, ii) the choice of model used by the initial NIDS is varied,

iii) the choice of XAI technique is varied, and iv) we assess the ability of our methodology to generalise to unseen data distributions. The overall results indicate the efficacy of our methodology to produce significant improvement gains (i.e., up to +36% gains achieved across the considered metrics and datasets) that exceed those achieved by other state-of-the-art intrusion detection models from the literature.

Chapter 6 - Conclusion & Future Directions - In this chapter, we detail the conclusions of this thesis and discuss some of the open and existing research questions relevant to our work. Finally, we elaborate on the future directions of this research.

1.6 Dissemination

This section details the publications undertaken to disseminate the research findings of this PhD project. Specifically, we separate journal and conference publications and associate each publication to the research question(s) that it addresses.

Journal

- **RQ1 and RQ3:** P. Barnard, N. Marchetti, and L. A DaSilva, "Robust network intrusion detection through explainable artificial intelligence (XAI)", *IEEE Networking Letters, June 2022*.

- **RQ1 and RQ4:** P. Barnard, L. A DaSilva, and N. Marchetti, "Don't Just Explain, Enhance! Using Explainable Artificial Intelligence (XAI) to Automatically Improve Network Intrusion Detection", under review, submitted to *IEEE/ACM Transactions on Networking*.

Conference

- **RQ1 and RQ2:** J-B Monteil, J. Hribar, P. Barnard, Y. Li, and L. A. Da Silva, "Resource reservation within sliced 5G networks: A cost-reduction strategy for service providers", *IEEE International Conference on Communications Workshops (ICC Workshops), June 2022*.

  My role in this paper included background research on related works, aiding the analysis of the results, writing parts of the abstract, introduction, and validation section. I also helped with the creation of figures used to help visualise the system model.

- **RQ1 and RQ2:** P. Barnard, I Macaluso, N. Marchetti, and L. A DaSilva, "Resource reservation in sliced networks: an explainable artificial intelligence (XAI) approach", *IEEE International Conference on Communications (ICC), May 2022*.

Chapter 2

Background Theory & Review of Related Work

In the previous chapter, we briefly introduced the concept of XAI and discussed the role that this emerging technology is expected to play in the standardisation of future telecommunication networks, which are increasingly driven by AI/ML algorithms. We also outlined the key research questions addressed in this thesis and mapped each research question to the contributions made across two specific use cases, namely resource management within a sliced network and intrusion detection within cybersecurity. The purpose of this chapter is twofold: in the first section, we introduce and outline the necessary theoretical background, tools, and frameworks required to contextualise the solutions presented in this thesis. This includes an overview of the primary AI/ML algorithms (Section 2.1.1) and evaluation metrics (Section 2.1.2) used in our work, an introduction to the general XAI taxonomy (Section 2.1.3) and specific XAI tools employed (Section 2.1.4), an overview of XAI evaluation methods (Section 2.1.5), and a brief discussion of related XAI topics and concepts (Section 2.1.6). The second section of this chapter reviews the relevant literature, starting with an exploration of explainability within the broader telecommunication domain (Section 2.2). We then narrow our focus to the specific gaps and challenges this thesis addresses within the contexts of resource management within sliced networks (Section 2.2.1) and intrusion detection within network cybersecurity (Section 2.2.2).

2.1 Technical Background

To understand the solutions presented in this thesis, this section provides a technical review of the main concepts explored in this thesis. Specifically, we begin this section by discussing the primary ML/AI techniques employed in our work (Section 2.1.1). We then introduce and discuss the general XAI taxonomy and related concepts (Section 2.1.3), followed by an in-depth discussion on the specific XAI methodologies used in our work (Section 2.1.4), as well as the various XAI evaluation metrics that we consider (Section 2.1.5). Finally, we conclude this section with a brief discussion on some of the main topics that closely relate to the broader field of XAI (Section 2.1.6).

2.1.1 AI/ML Techniques for Resource Management and Network Intrusion Detection

The term AI can be described as the "study of agents that receive precepts from the environment and perform actions" while the concept of ML forms a "subfield of AI that studies the ability to improve performance based on experience" [36]. In this thesis, we use the terms AI and ML interchangeably when referring to any model or algorithm that can be trained to perform some particular task, T. Generally speaking, an AI/ML model may be trained to perform either the task of classification or regression [37]. In the classification setting, the model is trained to map an input sample to a specific category or class (or the probability of belonging to that class), while, in the regression setting, the model is trained to map the input sample to a particular number on the real line. For example, in a resource management setting, a typical task is that of traffic prediction, where regression-based ML is predominantly employed to predict future traffic demands at a particular node within the network so that operators can better optimise the resources allocated to that node, such as prioritising bandwidth resources for a node serving video services and prioritising queuing resources for nodes serving emergency calls, etc. In cybersecurity, and in particular, the intrusion detection setting, classification models constitute one of the most common approaches used by cybersecurity analysts to analyse patterns of traffic behaviour and determine the probability of an attack occurring within the network.

In this thesis, we present solutions based on both classification and regression tasks. Throughout our work, we use lowercase letters, such as $f(.)$ or $g(.)$ to denote the underlying function implemented by a particular model or algorithm. We also use the vector representation $\mathbf{x} = [x_1, ..., x_i, ..., x_N]$ to denote a single input sample passed to the model, where x_i represents the i^{th} feature in $\mathbf{x}$, with $\hat{y}$ further used to denote the output of the function call, $f(\mathbf{x})$, which aims to approximate the truth label, y. Moreover, we use the notation $\mathbf{x} \in X$ to denote the empirical distribution, or dataset, from which sample $\mathbf{x}$ is drawn from, and correspondingly use, $y \in Y$ to denote the empirical distribution of the label set, from which y is drawn from. Finally, we consider numerous types of ML models throughout our work. Below, we briefly introduce and outline the basic operation of the main ML models considered in our work:

Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting, or XGBoost [38], is a prominent ML algorithm that incorporates an ensemble of weaker learners into its overall structure to produce a more powerful ML model. These weaker learners, which often comprise shallow decision trees, are combined iteratively into the overall structure of the XGBoost model, such that each newly added learner is optimised in reducing the errors made by the previous ensemble of learners. This approach, known as gradient boosting, allows XGBoost to efficiently leverage the strengths of each individual learner, resulting in significantly improved performance compared to that of each individual learner in isolation. Moreover, beyond its ensemble approach, XGBoost also incorporates different regularization techniques that further help the algorithm to avoid overfitting during its training and improve its overall performance. These factors allow XGBoost to excel at various tasks where structured data can be organised into a tabular-tyle dataset. In the field of telecommunications, XGBoost models have found tremendous success in numerous applications, including predicting customer churn, predictive maintenance, and network intrusion detection, etc. However, while its ensemble approach helps improve its performance, it also comes at the cost of reducing the model's overall level of transparency as the number of weak learners increases, thereby making it more challenging for humans to understand the internal decision-making process of this algorithm.

In our work, we utilise an XGBoost model primarily for performing network intrusion detection, where the output of the model essentially corresponds to the predicted probability of an attack occurring within the network.

Deep Neural Network (DNN)

Inspired by the biological structure and function of the human brain, a deep neural network, or DNN, refers to a type of Artificial Neural Network (ANN) where hundreds to thousands of interconnected processing units, also known as neurons or nodes, are used to perform the task of regression or classification. Specifically, within the architecture of the model, these neurons are arranged into layers, such that each layer receives inputs from the neurons of the previous layer. In contrast to other types of shallow ANNs, such as the Multilayer Perceptron (MLP), which typically comprises an input layer, a single hidden layer and a final output layer, DNNs make use of multiple hidden layers (i.e., typically three or more). This added benefit of employing multiple hidden layers has long been known to improve the overall expressiveness of the model [39], while more recent works have visually demonstrated the ability of hidden layers to progressively extract higher-order features and learn increasingly complex relationships from the input data [40]. Formally, let

$$f^{(i)} = \alpha^{(i)}(\mathbf{W}^{(i)} f^{(i-1)} + \mathbf{b}^{(i)}) \tag{2.1}$$

denote the output of the i^{th} hidden layer of a DNN model, where $\alpha^{(i)}$ corresponds to the activation function used by the i^{th} layer, $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ correspond to the weight matrix and bias vector, respectively, while $f^{(0)}$ corresponds to the input layer given by $\mathbf{x}$. Then the final output of a DNN with L hidden layers can be written as

$$f(\mathbf{x}) = f^{(L+1)} \circ f^{(L)} \circ f^{(L-1)} \circ \cdots f^{(1)}(\mathbf{x}) \tag{2.2}$$

Training of a DNN model generally involves the process of backpropagation [41], which is a type of optimisation technique whereby the weights and biases associated with each neuron of the DNN are iteratively adjusted to minimise the overall loss or error of the model during its training. By comparing the network's output with the desired target value, the backpropagation algorithm propagates this error signal backwards through

each layer, allowing the network to fine-tune its internal parameters and improve its performance. This capability to learn from data in a model-free manner has allowed the DNN to become a powerful tool in various tasks, such as image recognition, natural language processing, and time series forecasting.

In our work, we employ three main variations of DNNs, including

- **Vanilla DNN:** This is generally regarded as being the most common type of DNN design, whereby every neuron in a particular layer connects to all the inputs of each neuron in the next layer. This is also commonly referred to as a fully connected DNN.

- **LSTM:** This refers to a special type of recurrent neural network, in which the use of additional internal mechanisms (e.g., cell state memory, gates, hidden states) allows the model to more accurately learn long-term and time-dependent information from past data points in comparison to a vanilla DNN. This ability to retain relevant information over long time periods makes LSTM models a popular choice for tasks, such as series forecasting, natural language processing, and speech recognition, where context and order are crucial. In our work, we utilise both a vanilla DNN and LSTM model during the formulation of our first use case scenario, where the main objective is to learn how to make optimal reservation requests based on historical traffic demand data.

- **Deep Autoencoder:** An autoencoder is a type of DNN where the output of the model aims to replicate the input of the model in an unsupervised manner. A typical autoencoder consists of two main stages, namely an encoder stage, which compresses the input data into a lower-dimensional representation (latent space), and a decoder stage, which aims to reconstruct the compressed latent representation back into the original input data. In this sense, when an autoencoder is trained to accurately minimise the difference between the input data and the reconstructed output, the training process implicitly forces the encoder-decoder stages to automatically capture the most salient features of the training data [37]. In the general AI and ML literature, autoencoders have found widespread use in applications, such as dimensionality reduction, anomaly detection, and recommendation systems, etc. [42]. In our work, we employ a deep autoencoder to perform the task of

anomaly detection, where we treat any samples with a large reconstruction error as anomalous.

2.1.2 AI/ML Evaluation Metrics

In this subsection, we briefly summarise the various AI/ML evaluation metrics that we consider in our work when assessing the performance of our solutions developed within the resource management and network intrusion use cases.

AI/ML Metrics for Resource Management

We consider a number of metrics to assess the performance of our AI/ML solutions for resource management, including Root Mean-Square Error (RMSE), and over/under reservation, as defined by equations (2.3), (2.4) and (2.5), respectively. For each of these metrics, we note that $r^*_{t,k}$ refers to the optimal amount of resources required by tenant k at time t, $\hat{r}_{t,k}$ refers to the actual amount of resources reserved by tenant k at time t, while $t \in T$ refers to samples taken from the test set and $\mathbb{1}\{\cdot\}$ denotes the indicator function.

- **RMSE:** This metric measures the average magnitude of the reservation errors incurred by the resource model. For this metric, a lower RMSE value indicates a more accurate reservation model, while a higher RMSE value indicates larger errors and a less accurate model.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{|T|} (\hat{r}_{t,k} - r^*_{t,k})^2}{|T|}} \tag{2.3}$$

- **Over-Reservation:** This metric measures the average amount of resources that are over-reserved by the model whenever over-reservations occur. For this metric, a higher over-reservation score relative to the optimal value of zero indicates a greater tendency for the model to overpay for unnecessary resources.

$$Over = \frac{\sum_{t=1}^{|T|} (\hat{r}_{t,k} - r^*_{t,k}) \mathbb{1}\{\hat{r}_{t,k} > r^*_{t,k}\}}{\sum_{t=1}^{|T|} \mathbb{1}\{\hat{r}_{t,k} > r^*_{t,k}\}} \tag{2.4}$$

- **Under-Reservation:** This metric measures the average amount of resources that are under-reserved by the model whenever under-reservations occur. For this metric, a more negative under-reservation score relative to the optimal value of zero indicates a greater tendency for the model to allocate fewer resources than required, potentially degrading the QoS for its end users.

$$Under = \frac{\sum_{t=1}^{|T|} (\hat{r}_{t,k} - r^*_{t,k}) \mathbb{1}\{\hat{r}_{t,k} < r^*_{t,k}\}}{\sum_{t=1}^{|T|} \mathbb{1}\{\hat{r}_{t,k} < r^*_{t,k}\}} \tag{2.5}$$

AI/ML Metrics for Network Intrusion Detection

We consider multiple common intrusion detection metrics to assess the performance of our AI/ML models used for network intrusion detection, including accuracy, recall, precision and F1-score. Adopting a conventional approach, we use the class labels 0 and 1 to denote benign and malicious traffic, respectively. We then define the true positive (TP) rate as the proportion of attacks that are correctly classified as attacks, the true negative (TN) rate as the proportion of benign samples correctly classified as benign, the false positive (FP) rate as the proportion of samples incorrectly classified as attacks, and the false negative (FN) rate as the proportion of samples incorrectly classified as being benign. Under these four basis terms, we then calculate each metric as follows:

- **Accuracy:** This measures the proportion of samples that have been correctly classified as a percentage of the total number of samples in the dataset.[1]

$$Accuracy\,(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{2.6}$$

- **Recall:** This measures the percentage of attack samples that have been correctly classified as an attack.[1]

$$Recall\,(\%) = \frac{TP}{TP + FN} \times 100\% \tag{2.7}$$

[1]Here, a value of 100% signifies a perfect score, while a value of 0% signifies the worst possible score.

- **Precision:** This measures the proportion of attacks that have been correctly classified as a percentage of all the samples classified as an attack.[2]

$$Precision\,(\%) = \frac{TP}{TP + FP} \times 100\% \tag{2.8}$$

- **F1-Score:** This metric aims to provide an overall "goodness" score for a classifier by taking the harmonic mean of recall and precision.[2]

$$F1 - Score\,(\%) = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \times 100\% \tag{2.9}$$

2.1.3 Overview of XAI Taxonomy

As discussed in the initial chapter of this thesis, the field of XAI entails the development of techniques and methodologies capable of explaining the behaviour of complex algorithms through human-understandable explanations. Within this broad definition, several finer distinctions have emerged in the literature in recent years based on how different XAI methodologies formulate their explanations and the assumptions they make when doing so. In this subsection, we explore and discuss some of these main distinctions, particularly as they pertain to the work and solutions presented in this thesis. To structure our discussion, we organise these distinctions under a general XAI taxonomy, as summarised in Figure 2.1. This taxonomy offers a general overview of the XAI landscape relevant to the scope of our work and has been adapted from similar taxonomies presented in the recent survey studies of [29] and [22]. By framing these concepts within the overarching theme of the "black-box" problem, we highlight the key components of this taxonomy as follows:

Interpretability versus Explainability

One of the most fundamental distinctions often made in the field of XAI relates to the general approach taken when designing a solution to tackle a particular problem. In this respect, two main distinctions that dominate the overall XAI landscape, include those

[2]Here, a value of 100% signifies a perfect score, while a value of 0% signifies the worst possible score.

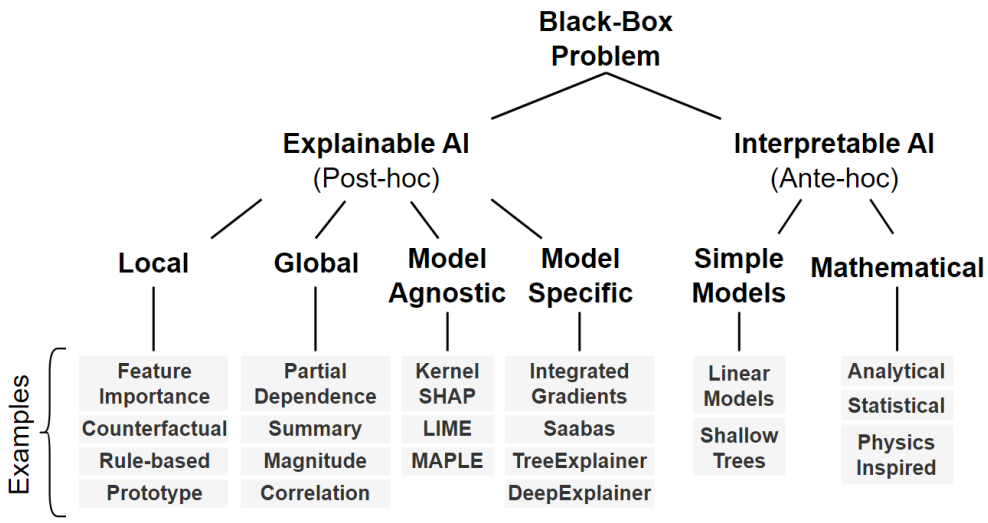


FIGURE 2.1: Overview of the XAI taxonomy adopted throughout this work.

that employ "interpretable AI" solutions and those that incorporate "explainable AI" in their design.

In this sense, interpretable solutions, also commonly known as *ante-hoc* explainability [43], refer to any solutions that aim to avoid the use of complicated black-box models in their construction. Instead, solutions in this category tend to make use of models that convey a simplistic architecture in their design, thereby allowing the decisions of the model to be intrinsically interpretable and easy to interpret by humans. For example, prior works in this category have proposed the weights of a sparse linear regression model (e.g., [44]) or rules of a shallow decision tree (e.g., [45]) to be interpretable. Additionally, a growing number of works have proposed the idea of encoding domain knowledge via mathematical representations into their solutions to provide interpretability. This includes works that make use of analytical expressions (e.g., [46]), statistical modelling techniques (e.g., [47]), or physics-informed solutions (e.g., [48]).

In contrast to interpretable solutions, explainable methods operate on the premise that the behaviour of a complex model cannot be understood simply by examining its internal structure and composition. Instead, methods from this class tend to rely on various *post-hoc* analysis techniques to systematically interact with the trained model and generate explanations that can be scrutinised by the HITL to gain insights into the model's

behaviour. Within this space, a number of explainable methodologies have been developed in recent years drawing inspiration from fundamental concepts such as sensitivity analysis (e.g., [49]–[51]), feature permutation (e.g., [52]–[54]), data visualisation (e.g., [55]), gradient analysis (e.g., [56]–[58]), and training interpretable surrogate models to approximate the behaviour of black-box models (e.g., [59], [60]).

Importantly, in this thesis, we focus primarily on post-hoc XAI methods, as we believe solutions of this form offer some key benefits to the telecommunication domain over their interpretable counterparts. Specifically, and while not always a clear-cut argument (for example, see the discussion in [61]), intrinsically interpretable models, such as shallow trees, linear models, etc., tend to be limited in their capacity and expressiveness to learn and extract complex patterns in their training data [62], [63]. Consequently, we argue that this limitation can potentially diminish the ability of these methods to deliver adequate performance to some of the complex scenarios that future networks are expected to present. On the other hand, XAI techniques, by definition, do not impose any restrictions on the complexity of the model or algorithm that they aim to explain (i.e. since they avoid compromising the accuracy of the model). We believe this characteristic makes XAI methodologies more suited towards the types of complex AI/ML algorithms regularly being pursued in the literature to combat the challenges faced by future networks. In the discussions that follow, we provide in-depth examples of the various types of explainable approaches that form part of the wider XAI taxonomy.

Local versus Global XAI

Another important and common distinction relates to the overall scope of the behaviour that the explanations try to capture and convey to the HITL. From this perspective, explanations can be classified as being either local or global, where

- **Local Explanations:** This refers to explanations that try to explain the behaviour of the model around a single output prediction of the model, i.e., which features in $\mathbf{x}$ are most important towards the outcome $y = f(\mathbf{x})$.
- **Global Explanations:** This refers to explanations that try to capture the general behaviour of the model across a wider context, such as the entire conditional distribution function learnt by the trained algorithm. We note, that in practice, many

global explanation methodologies operate by first generating a large set of local explanations, i.e., such as explaining the full training set, etc., and then aggregating this set of local explanations into a more compact representation that can be easily digested by the HITL, ie., computing the average importance of each feature.

Model Agnostic XAI versus Model-Specific XAI

Explanations can also be classified depending on whether the XAI methodology that produces them is model-agnostic or model-specific. In this sense,

- **Model-agnostic:** This refers to XAI methodologies that can explain a black-box model independent of the type of AI/ML model being explained. Typically, methodologies of this type only require access to the input and output of the model being explained, while the internal structure of the model is treated entirely as a black box.

- **Model-specific:** This refers to XAI methodologies that require access to the internal structure of the model in order to understand how it works. Importantly, while model-agnostic methodologies offer the advantage of being applicable to almost any model type, they also tend to require significantly more time to compute explanations in comparison to model-specific methodologies, which are generally able to take advantage of the model's internal structure to compute explanations at significantly less time and/or with greater accuracy.

Types of Questions Addressed

Finally, one other important distinction that can be made relates to the types of questions that each XAI methodology is able to answer regarding the behaviour of a black-box model. Some of the main distinctions of this category include:

- **Feature Importance/Attribution**: Methods of this kind aim to explain the behaviour of a model by identifying and ranking the impact of each input feature as it influences the output of the model. Methods of this particular class can be used to answer questions, such as "why" and "by how much" has each feature influenced the output of the model.

- **Counterfactuals:** Methods from this class typically formulate their explanations based on the minimal set of changes required to the input features in order to arrive at a particular output of the model. Explanations from this class can be used to answer "what-if" questions.

- **Prototypes:** These include representative instances of each class that aid in visualising and understanding the decision boundaries of the model. They provide insights into what the model considers to be a typical example for a particular class of interest. They can be used to answer questions such as, "what features are distinctively important for this class?" or "what makes this class different from that one?", etc.

In this thesis, we make extensive use of feature importance-based XAI techniques, encompassing both local and global scopes of explainability, as well as model-agnostic and model-specific methodologies. Moreover, while we acknowledge that other forms of explainability, such as prototypes, rule-based, and counterfactuals[3], etc., may also serve as potentially viable alternatives to the solutions proposed in this work, we recognise their absence as a current limitation of our research and a potential direction for future investigation.

Nonetheless, we highlight some of the main benefits associated with the adoption of feature importance-based techniques in the context of our work. Namely, following an in-depth literature review on the use of XAI techniques within telecommunications (we refer the reader to Section 2.2 for more details), we observe that feature attribution methods represent one of the most prominent and widely accepted forms of XAI adopted within this domain. In the context of our work, we believe this widespread prominence fosters greater accessibility to our findings while ensuring our research remains relevant to an established community of researchers in telecommunications.

Additionally, while there has been growing interest in alternative XAI approaches, such as counterfactuals, in recent years, our literature review on XAI evaluation methodologies (we refer the reader to Section 2.1.5 for more details) indicates that feature importance-based techniques benefit from a more mature and well-defined foundation of existing

[3]We consider counterfactual explanations to a limited extent in Section 5.3

research for assessing explanation quality. In contrast, evaluating the robustness of alternative approaches, such as counterfactuals, appears to remain an active area of investigation, with some recent studies highlighting potential vulnerabilities in these types of explanations, such as having potential susceptibility to spurious correlations in the model's behaviour [64]. Consequently, we argue that while the field of XAI continues to evolve, feature attribution methods currently provide a robust and interpretable foundation for exploring model behaviour within the context of our research. In the following subsection, we introduce and discuss in greater detail the various XAI methodologies relevant to our work.

2.1.4 XAI Techniques for Resource Management and Network Intrusion Detection

In this subsection, we delve further into the specific XAI methodologies, tools and concepts used in our work. In particular, we begin this subsection by introducing and defining a special class of local feature importance techniques, known as Additive Feature Attribution (AFA) methods. Importantly, methods from this class have previously been shown to produce intuitive and human-friendly explanations [54] and represent one of the most widely adopted classes of XAI techniques throughout the literature. Following this, we then introduce and discuss the various XAI methodologies and related tools that we employ throughout the later sections of this thesis.

Additive Feature Attribution (AFA) Explanations

Within the scope of local feature importance, one prominent and intuitive set of XAI techniques stems from the class of AFA methods. Essentially, methods from this class try to explain the output of a complex model $f(\mathbf{x})$ by fitting a linear model $g(\mathbf{x})$ around a specified region of the model space and subsequently using the regressed weights of the linear model to attribute importance scores ϕ_i among each of the input features based on the relative importance that each feature has on the output of the model:

Definition 1 (Additive Feature Attribution Methods)**.**
Let $f : X \rightarrow \mathbb{R}^1$ be a complex function that maps an N-dimensional input vector, $\mathbf{x}$ =

$[x_1, ...x_N] \in X$, to a real-valued output $f(\mathbf{x})$. For some input $\mathbf{x}$, AFA methods produce an explanation model $g(\mathbf{x})$, whose outputs comprise of a series of attributions, $\boldsymbol{\phi} = [\phi_1, ...\phi_N]$, corresponding to the effect that each input feature has on $f(\mathbf{x})$, such that the following holds true:

$$g(\mathbf{x}) = \phi_0 + \sum_{i=1}^{N} \phi_i \approx f(\mathbf{x}) \tag{2.10}$$

Intuitively, the functional form of this class means that the individual effects can be summed up to approximate the *difference* between the model output and a baseline effect, ϕ_0. Typically, this baseline is chosen to correspond to an uninformative aspect of the model space so that the linear model is used to explain *why* the output of the model differs from this baseline. For example, a black region is typically used in image processing applications while an empty string may be used in text analysis problems, etc.

Importantly, we note that a vast number of techniques exist within the XAI literature which conform to the class of AFA, for example, see the discussion in [65]. In this thesis, we make use of several different AFA-based XAI methodologies to help develop and evaluate our proposed solutions. Below, we discuss each of these methods in greater detail:

SHAP Values

Throughout this thesis, we make extensive use of the SHapley Additive exPlanation (SHAP) methodology proposed by [54]. This particular methodology makes use of the Shapley values from game theory to construct explanations that have a strong mathematical underpinning and exhibit a number of desirable properties. We note that this method has become one of the most widely adopted approaches for feature importance due to its provably unique and desirable properties, which lack across most other XAI methodologies proposed to date. To understand the basic principle of this methodology, we briefly review its association with the Shapley values. Specifically, the Shapley value defines a unique allocation rule for distributing surplus credit among each player in a coalitional game according to their individual contributions to the grand coalition. Formally, the Shapley value for player i in an N-person game can be defined as:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N|-|S|-1)!}{|N|!}(v(S \cup \{i\}) - v(S)), \tag{2.11}$$

where $S \in \mathcal{P}(N)$ refers to a subset of players from the power set of N, and $v(S) \in \mathbb{R}^1$ denotes the payoff that is obtained following the outcome of a game with players in S.

Intuitively, equation (2.11) shows that the Shapley value for player i is obtained by observing the player's marginal contribution to the payoff when joining a particular coalition of players, i.e., the change in payoff that occurs when the player is included in a game ($v(S \cup i)$) versus the payoff that occurs when the player is excluded from the game ($v(S)$), and taking a weighted average across all $|N|!$ possible coalitions that can be formed from the grand coalition of players in N.

In the field of XAI, the authors of [53] and [66] have recently demonstrated the use of Shapley values as a local additive attribution method by interpreting the model as a game, its inputs as players and output as the payoff function. In this sense, the Shapley values present a particularly promising approach to assigning importance amongst each of the input features of a black-box model as it is provably the only credit distribution that guarantees a number of fairness-related properties [67]. In the context of XAI, these properties present a strong theoretical justification for their preferred use over many other methods in the literature, which fail to uphold these same desirable properties:

- **Local Accuracy:** This property guarantees that explanations based on the Shapley values will precisely sum to the difference between the model output and the baseline value ϕ_0.
- **Consistency:** This property enforces monotonicity in the sense that a feature whose impact increases over time cannot receive a decreasing attribution.
- **Missingness (Dummy):** This property guarantees that a feature which does not contribute to any of the coalitions receives an attribution score of zero.

In SHAP, the authors define the payoff function for a subset of features as the expectation of the model output conditioned on the features which are present in the subset, i.e.,

$$v(S) = E[f(\mathbf{x})|\mathbf{x}_S], \tag{2.12}$$

with $\mathbf{x}_S$ being the set of inputs that are present in a particular coalition. However, in practice, as most AI models cannot cope with 'missing' inputs, the authors propose to approximate this conditional by assuming feature independence and model linearity to give

$$v(S) = E[f(\mathbf{x})|\mathbf{x}_S] \approx f(x_s, E[x_{\bar{s}}]) \tag{2.13}$$

where $\bar{s}$ refers to the set of inputs which are 'missing' in the coalition. Moreover, the authors propose to define the baseline effect as the expected model output when all inputs are present, i.e., $\phi_0 = E[f(\mathbf{x})]$, and use empirical sampling from a background set of input samples to calculate the expectation of the model's output when a subset of features are present. Fundamentally this means that the importance scores, ϕ_i, can be interpreted as *the change in the expected output of the model when that feature's value is known to the model versus when it is unknown to the model*, which the authors refer to as SHAP values.

Throughout this thesis, we make use of three open-source implementations provided by the original authors in [54] to compute SHAP explanations as part of our proposed solutions, including the 'Kernel SHAP', 'TreeExplainer' and 'DeepExplainer' methods. In particular, we note that the 'Kernel SHAP' method presents a model-agnostic approach that can be used to compute exact SHAP values of any black-box algorithm when the input space is relatively low (i.e., typically less than 20 input features), and sparse approximations otherwise. On the other hand, the 'TreeExplaine' method is a model-specific approach that can be used to compute exact SHAP values for tree-based models within polynomial time complexity. Finally, the 'DeepExplainer' is another model-specific approach that can be used to compute SHAP values specifically for neural-based models.

LIME

Originally proposed by [59], the LIME algorithm provides a model-agnostic approach to explain the individual predictions of any black-box model based on the impact of its features. As the main intuition behind this approach, LIME trains an interpretable surrogate model, such as a sparse linear model, to approximate the behaviour of the original model around a neighbourhood centred at the sample to explain, and then uses the trained weights of the surrogate model to indicate which features are important to the model for

that sample. Throughout this thesis, we make use of the open-source implementation provided by [59] to compute LIME explanations as part of our proposed solutions.

Saabas

Proposed by [68], the Saabas method refers to a feature attribution-based approach that has been designed specifically to explain the behaviour of tree-based models. In this method, feature importance is calculated by considering the decision path responsible for producing a particular prediction of the model, and attributing any changes in the model's expected output among each of the features used along this path. Throughout this thesis, we use the open-source implementation, 'TreeInterpreter' [69], to compute Saabas explanations as part of our tree-based solutions.

MAPLE

Another feature attribution-based approach considered in our work includes the recently proposed Model Agnostic Supervised Local Explanation (MAPLE) explainer [70]. Similar to LIME, this method also builds upon the idea of utilising simplified linear surrogate models to approximate and explain the behaviour of any black-box model around a particular input of interest. In addition to this, the authors also propose the use of additional decision trees to learn a weighting distribution across the training data. This in turn allows the explanations to implicitly detect global patterns while providing more faithful explanations compared to LIME. In our work, we use the open-source implementation of MAPLE provided by [70] to construct explanations of some of our proposed solutions.

Integrated Gradients

In addition to the above explanation techniques, we also consider the Integrated Gradients (IG) method from [71] in our work. As suggested by its name, this method operates by assigning importance scores to each of the input features based on their partial gradient with respect to the output of the black-box model being explained. In comparison to other gradient-based approaches, this method includes an additional integration step that averages gradients across multiple versions of the original input scaled against a reference baseline. As shown by the original authors in [71], this simple but important step

can significantly enhance the perceived quality of the explanations by avoiding problems related to gradient saturation (for an in-depth discussion on this topic, we refer the reader to [72]). We note that this method can only be applied to models that have a continuously differentiable gradient, i.e., can be applied to neural network-based models but not tree-based models, which instead learn a non-differentiable function based on discrete decision rules. Throughout this thesis, we apply the IG approach to some of the DNN-based solutions that we propose.

2.1.5 XAI Evaluation Methods

For almost as long as explanation methods have been proposed to explain the predictions of AI/ML models so have there been approaches proposed that aim to evaluate the quality of the resulting explanations. Generally speaking, the literature on this topic can be classified along two main directions, where one set of approaches relies on human subject studies to qualitatively evaluate the quality and benefits of explanations, while a second and more recent set of approaches aims to evaluate the quality of the explanations in a strictly objective and quantitative manner. In this subsection, we briefly review and summarise the literature under both of these main groups and then outline and discuss the specific XAI evaluation metrics that are considered in this thesis.

Literature Review

Over the past two decades, numerous evaluation methodologies have been proposed in the literature to assess various quality-related factors of AI/ML model explanations. Early approaches often relied on carefully designed experiments involving human participants to evaluate qualitative properties such as user trust [73] and acceptance [74] of these explanations.

A notable example in this domain includes the early work of [75], which conducted a series of human-centred experiments to evaluate how well end users can understand and make actionable decisions based on the insights gained from different types of explanations for an email classification task. In their experiments, the authors constructed two separate ML models using the Ripper rule-learning algorithm [76] and the Naïve Bayes algorithm for email classification. After training both models to similar accuracy on the same dataset, three types of local explanations were generated across each test

sample, including rule-based explanations derived from the most dominant rule used by the Ripper algorithm, feature importance explanations based on the top positive and negative-weighted keywords/features used by the Naïve Bayes classifier, and prototype explanations based on the most similar samples from the training dataset to match each prediction in the test set.

In a series of rounds, participants were then given a chance to familiarise themselves with various combinations of input samples along with their corresponding classifications and explanations, after which two main experiments were conducted to measure the overall quality of the explanations as perceived by the users' feedback and insights. Specifically, in the first experiment, participants were given a series of email messages (i.e., input samples) without any accompanying classification labels or explanations and asked to classify each sample as well as to provide reasons for their classification based on the styles of the three explanation types. The results of this first experiment found that users were often able to produce explanations matching closely to those of the rule-based and feature importance based explanations. However, in the case of prototypes, it was found that users often found it difficult to identify similar samples to those identified in the prototype explanations.

In the second experiment, participants were given a series of samples with their corresponding classifications from each model and subsequently asked to explain why they believe the models made their decision, i.e., to reproduce what they believe the corresponding explanations should look like. Consistent with the first experiment, participants were able to produce explanations matching closely to those generated by the rule-based and keyword explanation types. On the other hand, participants demonstrated a reduced ability to identify similar prototypes, highlighting a limitation of this explanation type.

Crucially, the early work of [75] has laid the foundation for various subsequent studies that have since adopted similar variations of these two main experiments to qualitatively assess their explanations. For instance, the authors in [58] proposed a popular explanation method for Convolutional Neural Networks (CNNs), known as Gradient-weighted Class Activation Mapping (Grad-CAM). In order to validate their method, the authors conducted a human study to assess how well Grad-CAM explanations help users

build trust in predictions from DNNs. Specifically, in the first of two experiments, participants were shown explanations in the form of visual heatmaps and asked to identify which class they believed each explanation belonged to. By comparing the results across four other explanation methods, the authors found that participants successfully identified the correct class in 61.23% of cases when presented with Grad-Cam explanations, compared to only 44.44% achieved by the next best explanation method. In a second experiment, participants were then shown explanations from two models trained on the same task — with one model having higher accuracy than the other — and asked to identify which model they believed to be more reliable. Surprisingly, participants consistently identified the higher-accuracy model, supporting the authors' claims about the trustworthiness of Grad-CAM.

A similar user study was also conducted in [59] to validate the effectiveness of the LIME method. In this particular case, the authors used a dataset with known feature bias to train two models: the first model trained on the original dataset and the second model trained on a cleaned version with any biased features removed. Despite the second model achieving a lower test accuracy than the first, the authors found that users were still able to consistently identify the second model when shown explanations of both models and asked to identify the more reliable model. More recently, the authors of [54] conducted a user study to investigate and compare the benefits of SHAP against LIME and another popular AFA method known as DeepLift [72]. As part of their study, participants were shown examples of input-output pairs for different models and subsequently asked to assign feature importance under unseen inputs. The results of their study found that human-generated explanations tend to align more consistently to those generated by SHAP than in the case of LIME and DeepLift.

While there has been strong support for qualitative evaluation methods throughout the XAI literature, others have argued that methods for assessing the quality of explanations solely on visual inspection can be crucially misleading as this provides no true guarantee that the underlying explanations are in fact influenced by the parameters of the trained model or the training data [77]. Noticing a clear gap in the literature with regard to quantitative validation strategies, the authors in [78] proposed one of the earliest methodologies for objectively evaluating the quality of feature importance-based explanations for neural networks. Essentially, the proposed evaluation strategy in [78] consists

of first identifying the most important features in the original input as estimated by the magnitude of each feature's corresponding importance score in the explanation. Then, in steps of increasing fraction size, the most important features in the original input are "removed" from the input by replacing their original values with noise drawn from a uniform distribution. Under the assumption that features which are truly more important to the model are more likely to cause greater changes in the original prediction of the model when perturbed, the quality of the explanation is assessed by plotting the relative change in the model output (y-axis) versus the fraction of perturbed features (x-axis) and computing the area under this perturbation curve.

Importantly, by applying their proposed evaluation strategy to the problem of image classification and across various image datasets and explanation methods, the authors in [78] were able to conclude that the Layer-wise Relevance Propagation (LRP) explanation method provides quantitatively better explanations than two other explanation methods, namely a sensitivity-based XAI method proposed by [51] and a gradient-based XAI method proposed by [40]. This conclusion was made on the basis that the LRP method had achieved a greater area under the perturbation curve than those of the other explanation methods, suggesting that the LRP method was more accurate at identifying the most important pixels used by the image classifier.

Since the pioneering work of [78], there has been an increasing number of works adopting the general idea of perturbing or masking the most important features as indicated by an explanation and observing the change in model output to assess the quality of an explanation. In [72], the authors proposed a procedure in which the quality of an explanation method could be assessed by its ability to identify the most relevant features required to change the output of the model from its current prediction to that of a new target output. Demonstrating their approach to image classifiers, this procedure entails computing the element-wise difference between an explanation that has been computed with respect to the current predicted class and an explanation computed with respect to the target class (i.e., with the same input supplied to the model in both cases). Up to 20% of pixels with the largest positive difference are then masked by replacing their values in the original input image with noise. The quality of the explanation method is then assessed by observing the change in the log-odds score between the predicted class outputs of the original image and the modified image. By applying this procedure across various

explanation methods, the authors in [72] were able to conclude that DeepLift produces better quality explanations in comparison to some other popular gradient-based XAI approaches since it generally led to the largest change in prediction score towards the target class for an image classifier trained on the MNIST dataset [79].

Similarly, the authors in [80] conducted an experiment to determine which set of explanations derived from two main variations of the SHAP method could hypothetically provide applicants with the most useful insights required to improve their likelihood of receiving a bank loan following the output of a credit score model. In this experiment, the important scores of each explanation variant were first ranked in order of features identified as most likely to reduce the risk of not receiving a loan. In increasing step sizes, the most important features were then replaced using mean masking and the resulting change in the model output was observed. In this sense, the preferred explanation method was deemed to be the one which could improve the applicant's credit score the most following the least amount of masked features. More recently, the authors in [81] have expanded upon the general concepts explored in previous studies involving feature perturbation to evaluate explanation quality by introducing a comprehensive set of quantitative evaluation metrics. Essentially, these metrics involve a series of repeated feature perturbation experiments, where each repeated experiment aims to evaluate a specific quality of an explanation, such as its ability to correctly distinguish between features that increase/decrease the model's prediction and the overall importance ranking of each feature. We later describe these metrics in greater detail in the following subsection.

In this thesis, we primarily adopt quantitative approaches to assess the quality and faithfulness of our explanation-driven solutions. Similar to the arguments made in [77], [78], [80], we believe that assessing explanations exclusively on human inspection poses a potential risk of bias in the results as humans may favour explanations that resemble their own expectations while potentially penalising methods that in fact reflect the true behaviour of the model more accurately. Additionally, we emphasise that the solutions presented in this thesis are primarily designed to support additional ML/AI tools rather than being exclusively tailored for end users. For this reason, we argue that prioritising human reasoning or aligning with a predefined ground truth of what end users may find important in an explanation comes secondary to ensuring that explanations faithfully reflect the true underlying reasoning of the ML/AI classifier by accurately identifying

the features used by the classifier to support its decisions. Finally, while we acknowledge that the terms "quality" and "faithfulness" are often used interchangeably in the broader XAI evaluation literature, in this thesis, we make an important distinction between them. Namely, we refer to the former as the ability of one explanation method to score greater than another explanation method for a particular quality metric, while we purposely refer to the latter as the ability of an explanation method to score close to the optimal attainable value of a particular quality metric. We make this important distinction as we find that the vast majority of quality metrics proposed in the literature do not offer a ground truth score that allows the quality of the explanation method to be contextualised against the best-possible case. In the following subsection, we briefly review the XAI metrics that we consider in our work to evaluate the faithfulness of the explanations produced by our proposed solutions.

XAI Evaluation Metrics

We consider a number of XAI-specific metrics recently proposed in the work of [81] to evaluate the quality and faithfulness of our solutions. Essentially, these metrics can be used to cross-validate different explanation methods against one another (i.e., SHAP versus LIME, etc.) in their ability to more accurately identify different characteristics of the model's behaviour. Apart from the "Local Accuracy" metric discussed below, all remaining metrics from [81] lack an absolute scale of ground truth in their calculation, which becomes essential when evaluating the true faithfulness of the explanation. Below, we briefly summarise each of these metrics as they were originally presented in [81]. We later present in Chapter 3, our solution for computing the necessary ground-truths for each of these metrics in order to compute the true faithfulness of our explanations. For additional details on these metrics, we refer the reader to the supplementary material of [81].

- **Local Accuracy:** This metric measures the ability of an additive attribution method to estimate importance scores ϕ_i for each feature $i \in \{1, ..., N\}$ that, when combined with the baseline value ϕ_0, sum up to produce the original output of the model $f(\mathbf{x})$ for the sample being explained. Similar to [81], we evaluate the local accuracy of our proposed methodology by computing the normalized standard deviation of

the difference between the feature importance scores and model output across 100 random samples from the test set using equation (2.14):

$$\sigma = \frac{\sqrt{E_x[(f(\mathbf{x}) - \sum_{i=0}^{N} \phi_i)^2]}}{\sqrt{E_x[f(\mathbf{x})^2]}}. \tag{2.14}$$

To determine how well an explanation method performs on this metric, [81] proposes the use of nine cutoff regions within the range of 1.00 (highly accurate) and 0.00 (highly inaccurate) to rank the local accuracy σ of an explanation method:

$$\sigma < 10^{-6} \implies 1.00 \tag{2.15}$$

$$10^{-6} \leq \sigma < 0.01 \implies 0.90 \tag{2.16}$$

$$0.01 \leq \sigma < 0.05 \implies 0.75 \tag{2.17}$$

$$0.05 \leq \sigma < 0.10 \implies 0.60 \tag{2.18}$$

$$0.10 \leq \sigma < 0.20 \implies 0.40 \tag{2.19}$$

$$0.20 \leq \sigma < 0.30 \implies 0.30 \tag{2.20}$$

$$0.30 \leq \sigma < 0.50 \implies 0.20 \tag{2.21}$$

$$0.50 \leq \sigma < 0.70 \implies 0.10 \tag{2.22}$$

$$\sigma \geq 0.70 \implies 0.00 \tag{2.23}$$

- **Remove Positive (Mask):** This metric is designed to measure how well an explanation method can identify features that increase the output of the model the most. By iteratively removing increasing fractions of the top most features identified as having a positive effect on the model's output, i.e., by masking their values with their expected ones, the output of the model should be seen to decrease rapidly. Similar to [81], we compute this metric across 100 random samples from the test set. By plotting the fraction of features removed (x-axis) against the mean model output (y-axis), a corresponding curve is produced. To determine the performance of our proposed methodology on this metric, we calculate the area under this curve and compare it to the area produced by an optimal-achieving method. For this metric, a good explanation method should yield a low area under the curve.

- **Remove Negative (Mask):** This metric is designed to measure how well an explanation method can identify the features that decrease the output of the model the most [81]. Similar to the Remove Positive metric, this metric iteratively removes increasing fractions of the top most negative-effecting features, as identified by the explanation method. To determine the performance of our proposed methodology, we compute this metric across 100 test samples and compare the area under its curve to that of the optimal case. For this metric, a higher area under the curve implies greater performance.

- **Remove Absolute (Mask):** This metric is designed to measure how well an explanation method identifies the features that impact the accuracy of the model the most. Similar to the Remove Positive metric, this metric iteratively removes increasing fractions of the top most important features, where the importance of a feature is based on its absolute importance score and plotting the fraction of features removed (x-axis) against the mean change in model accuracy across multiple samples from the test set. For this metric, a higher area under the curve implies greater performance.

- **Keep Positive (Mask):** Similar to the Remove Positive metric, this metric also measures how well an explanation method can identify the features that increase the output of the model the most. However, unlike the Remove Positive metric, this metric begins by masking all features and then iteratively unmasking increasing fractions of the top most positive-effecting features and plotting the output of the model versus the fraction of unmasked features. For this metric, a higher area under the curve implies greater performance.

- **Keep Negative (Mask):** Like the Remove Negative metric, this metric also measures the ability of an explanation method to identify the features that decrease the output of the model the most. Unlike the Remove Negative metric, however, this metric starts by masking all features and then iteratively unmasking increasing fractions of the top most negatively affecting features. For this metric across, a lower area under the curve implies greater performance.

- **Keep Absolute (Mask):** Similar to the Remove Absolute metric, this metric also measures the ability of an explanation method to identify the features that impact the accuracy of the model the most. Unlike the Remove Absolute metric, however, this metric starts by masking all features and then iteratively unmasking increasing fractions of the top most important features. For this metric across, a lower area under the curve implies greater performance.

2.1.6 XAI Related Concepts

In this subsection, we briefly highlight some of the most common terms that often find discussion among the general topic of XAI in the literature:

Trustworthiness

The term trust or trustworthiness of an AI system has previously been described as "the set of mechanisms used in the explainable layers of the XAI model that enriches the learning model in order to be trusted by the users with a clear understanding" [82]. In particular, when an AI system operates as a black-box, this can leave stakeholders wondering how its decisions are made, leading to concerns about bias, manipulation, and unfairness. In this sense, explanations can foster greater trust in AI systems by allowing stakeholders to critically evaluate the actions of the system, which in turn can help users build greater trust in the system. At the same time, different stakeholders may also have different trust expectations. To help address this challenge, some researchers have recently proposed their own trust model that can allow explanations to be tailored to the specific context and audience to provide explanations that are relevant, understandable, and actionable for each group [21].

Causality

While XAI focuses on making AI models more interpretable and explaining their decisions, the concept of causality is more concerned with understanding the cause-and-effect relationships within a system. These concepts can be complementary, as incorporating causal reasoning into an AI model can contribute to its explainability by providing a

deeper understanding of why certain decisions are made [64]. Moreover, causality is typically relevant in situations where understanding the true impact of interventions, policy changes, or other actions is crucial, while XAI is generally more useful in scenarios where transparency, accountability, and user trust are important.

Fairness, Bias & Ethics

The term fairness has previously been described as "the absence of any prejudice or favouritism towards an individual or a group based on their inherent or acquired characteristics" [83]. At the same time, a system which exhibits fairness also necessarily exhibits desirable ethical characteristics, such as being free from any unwanted biases and discriminatory actions [83]. In the context of AI/ML, evaluating the fairness of a particular algorithm or model generally entails the process of ensuring that the probability of each outcome from the model or algorithm is independent of any sensitive features, such as gender, ethnicity, sexual orientation, or disability, etc. [84]. While various solutions have previously been proposed to evaluate the fairness of a system based on statistical or causal techniques, explanations have also shown promising results towards uncovering hidden biases and traits of unfairness in AI systems [85]. Moreover, the telecommunications industry, with its diverse demographics and socioeconomic groups, is particularly vulnerable to algorithmic bias. In this sense, XAI can help to uncover unfair biases within AI models, such as those leading to discriminatory pricing or prioritisation of certain demographics over others. By making these biases transparent to the HITL, stakeholders, such as regulators and model developers, etc., can proactively address them, ensuring their algorithms do not perpetuate any societal inequalities.

Accountability & Responsibility

XAI and accountability in AI are two important concepts that are intricately linked in their effort to foster greater responsibility and trustworthiness in AI algorithms. By explaining the factors that influence the decisions of an AI system, XAI helps to attribute responsibility to those involved in the algorithm's creation and deployment. Moreover, as explanations can also be used to audit an AI system for potential biases, errors, or unexpected behaviour, it thus becomes easier for stakeholders to identify and address issues that could lead to harmful or unfair outcomes. In this sense, understanding "why"

a system takes certain actions allows regulators to hold relevant actors accountable, encouraging responsible development and deployment of AI within the telecom industry [86].

2.2 Explainability/Interpretability in Telecommunications

Despite its relative infancy as a research area, the general topic of explainability and interpretability has received remarkable interest from the telecommunications community in recent years. The survey papers by [82], [87], [88] highlight just some of the many areas where researchers in the telecommunications domain have already applied solutions based around the need for greater explainability and interpretability in the use of complex algorithms. Below, we discuss some of these works and also attempt to cluster them into the main problem areas to which they have been applied to.

One of the most prominent areas of research where explainable and interpretable AI has flourished within the telecommunication domain in recent years is optimisation at the physical layer. For example, in the work of [89], the authors propose an interpretable human intelligence-guided AI-enhanced constellation design method for Beyond 5G (B5G) Non-Orthogonal Multiple Access (NOMA) systems. As part of their solution, the authors propose to leverage expert domain knowledge to design and optimise two "mini neuron networks" to perform the tasks of modulation and power allocation separately before combining the outputs of each model into their overall solution. The authors demonstrate that this technique drastically lowers the amount of training data required to train the entire system compared to if only a single End-to-End (E2E) neural network were to be employed, while they claim the use of smaller neural networks also serves to make their overall solution more interpretable and transparent.

This general sentiment of decomposing a particular problem into two or more smaller problems that can be solved separately using shallow ML techniques and subsequently combined to enhance the transparency of the overall system (i.e., compared to an E2E approach) has also been shared by the authors in [90]. In this work, the authors investigate the problem of link adaption within wireless networks, where the main objective is to maintain high spectral efficiency by dynamically adjusting link transmission parameters, such as the Modulation and Coding Scheme (MCS), in response to the dynamic

and fast-changing conditions of the network channel. To this end, the authors propose and compare two approaches based on deep learning that can be used to perform link adaption. The first of these includes an E2E solution, where a neural network is trained directly on historical Channel State Information (CSI) to predict the optimal MCS that should be employed at the next time slot. The second approach, referred to as a "hybrid" approach, first computes a sufficient statistic of lower dimension based on the historical CSI, and is then fed as an input feature into a smaller-sized neural network to predict the optimal MCS. While the authors provide no tangible examples (i.e., such as explanations) that allow the behaviour of their models to be understood, a similar argument is made to that of [89], whereby the latter solution is viewed as being more interpretable due to the use of a smaller-sized neural network.

Also on the topic of interpretable NOMA, the authors in [91], have recently proposed a hybrid-cascaded DNN architecture for cooperative NOMA, which incorporates both multiple loss functions to better quantify the Bit Error Rate (BER) of the system as well as a multi-task orientated two-stage training strategy that allows their overall system to be trained in an E2E manner. To help demystify and verify the learning mechanism of their DNN-based solution, the authors propose to use information theory to map the loss functions of their DNN solution into probability distributions, which can then be visually compared against the output distributions of their trained solution for consistency.

While the above-mentioned studies have mainly considered full E2E solutions at the physical layer, others have focused primarily on addressing specific aspects within this overall problem space. For example, the problem of power control has by itself received bounteous attention from researchers aiming to improve the interpretability of their solutions. In [46], the authors demonstrate that many Radio Resource Management (RRM) problems can be formulated as a graph optimisation problem and propose the use of Message-Passing Graph Neural Networks (MPGNNs) to solve tasks, such as power control and beamforming. By proving the equivalence between MPGNNs and the family of distributed logical iterative optimisation algorithms (i.e., such as the classical Weighted Minimum Mean-Square Error (WMMSE) algorithm used in power allocation), the authors are able to analytically provide tractable guarantees about the performance of their solutions, which they further argue helps to improve the interpretability of their solution.

Meanwhile, in the work of [92], the authors propose an interpretable AI solution for

power steering and control of intelligent reflective surfaces. Essentially, their proposed approach involves mapping each tile of the intelligent surface to a unique node within a neural network, while the wireless propagation paths (i.e., line-of-sight connectivity between tiles) are mapped to the individual links and their weights of the neural network. The authors argue that this approach improves the interpretability of their solution since the floor plan geometry of the reflective surface becomes directly embedded into the structure of the neural network (i.e., physics-inspired architecture).

In [93], the authors present some initial results for explaining the underlying function learnt by a neural network for power allocation of ODFM communication channels. Their approach begins by first training a shallow two-layer neural network that approximates the classic water-filling algorithm for power allocation within a simulated network environment. To explain the high-dimensional behaviour of their model, the authors propose to fit an interpretable surrogate model based on the Mejer G-function [94] that approximates the decisions of the neural network. While the Mejer G-function essentially serves as a low-dimensional symbolic mapping of the neural network's behaviour, the authors acknowledge there is still no guarantee that the surrogate model has learnt the same underlying function as the neural network, a problem which they intend to address in future work.

In addition to the problem of power allocation, many papers have focused primarily on the problem of signal interference and classification. For example, in [95], the authors discuss the importance and general lack of interpretable models available for interference detection of wireless signals. To tackle this problem, the authors propose their own interpretable solution for interference detection of Wireless Fidelity (WiFi) signals. Their proposed approach makes use of a non-negative multiple linear regression model to perform interference detection, where the weights of their trained model correspond to the estimated power levels of each transmitting device within the network. By performing an experiment where two devices transmit simultaneously, the authors demonstrate that their interpretable model is comparably accurate as a DNN-based solution for interference detection and that their model intuitively considers the distance between the two devices as part of its detection process.

In [96], the authors employ the Grad-CAM XAI methodology to investigate how different radio features affect the classification performance of neural networks for signal

modulation classification. Essentially, the Grad-CAM algorithm operates by estimating the partial gradient of the input features (i.e., in this case, the inputs consist of 2-dimensional in-phase and quadrature (I/Q) samples) with respect to the output(s) of the neural network. By employing the Grad-CAM method, the authors are able to create explanations in the form of heatmaps, which demonstrate and reveal how different areas of the input space tend to be more important for the neural network in distinguishing between different types of modulation schemes, such as Quadrature Phase-Shift Keying (QPSK), Wide Band Frequency Modulation (WBFM), Quadrature Amplitude Modulation (QAM), Gaussian Frequency Shift Keying (GFSK), etc.

In [97], the authors aim to determine the variables which have the most significant impact on WiFi throughout. To achieve their goal, the authors first propose to train two types of ML models, namely a linear regression model and a random forest model, to predict WiFi throughout based on input variables, such as link speed, received signal strength, Round-Trip Time (RRT), and the number of Access Points (APs) available within the network, etc. To determine the significance of each variable, the authors propose of number of conventional approaches for data and feature analysis, including pairwise correlation and the Percentage of Variance (PoV) explained by each variable on the overall performance of each model. Their analysis reveals that while the regression model does not appear to significantly favour any single variable more than the remaining ones, the random forest model, in contrast, relies heavily on the RRT to achieve high predictive performance.

In [98], the authors present a preliminary theoretical analysis to interpret the performance tradeoffs of deep learning solutions that make use of the Rectified Linear Unit (ReLu) activation function at their output layer for channel estimation. In particular, the authors proposed approach makes use of statistical learning theory to first prove an equivalence between fully-connected ReLu DNNs and piecewise linear functions. Based on this equivalence, the authors then demonstrate that ReLu DNNs also serve as effective universal approximators for conventional signal processing methods used for channel estimation, such as the Minimum Mean-Squared Error (MMSE) estimator in particular. While the results of their analysis do not go as far as to offer tangible explanations describing the behaviour of ReLu DNNs for channel estimation, the authors argue that it

does provide an initial theoretical stepping stone towards ensuring performance guarantees for ReLu-based deep learning solutions for channel estimation.

In [99], the authors consider the problem of nonlinear interference mitigation via the equalization of nonlinear losses of a communication channel. While this problem can typically be solved using a traditional approach that uses digital back-propagation methods, such as the split-step Fourier method (SSFM), the authors instead propose to design a physics-informed DNN solution for this problem, which essentially approximates this process in the form of learned digital back-propagation (LDBP) by unrolling the SSFM iterations and approximating each span inversion with two neuron layers. Importantly, while this direct mapping can help improve the overall interpretability of the model, its application is generally limited to problems where known mathematical models already exist while the overall achieved performance is also typically lower than that of a full-scale deep learning approach.

While most of the studies discussed so far have focused on the role and benefits that XAI and interpretable AI methodologies can offer to problems occurring at the physical layer, other works have also considered the benefits that these methodologies can offer at higher levels of the network stack, such as understanding the factors that impact overall network service quality. For example, in [100], the authors aim to devise a solution which is both interpretable and can accurately predict the Quality of Experience (QoE) perceived by end customers within a wireless network. To achieve these goals, the authors propose the use of causal inference based on the Linear Mixed Models (LLM) method to identify and isolate the key features that have a direct causal impact on the QoE of customers from random features, also known as confounders. As part of their evaluation, the authors apply their solution to a dataset comprising both non-causal customer data, such as gender, age, user level, monthly Internet fee, etc., as well as conventional network quality indicator data which they define as being causally-related to the QoE of users. The authors demonstrate that their proposed approach is more accurate at predicting QoE in comparison to traditional methods, such as the Pearson correlation, linear regression, etc., while its ability to identify causal features makes their solution more interpretable than conventional ML approaches.

In [101], the authors propose a novel methodology for explaining the results of unsupervised ML methods for clustering. Their proposed approach referred to as "EXPLAIN-IT," works by first applying any black-box clustering technique to a dataset to form a group of clusters. In order to explain how each cluster has been formed, the authors then propose to model the results of the clustering algorithm using a supervised ML approach with strong discriminatory power, such as a Support-Vector Machine (SVM) model, for instance, and subsequently employing post-hoc local XAI methodologies, such as Local Interpretable Model-agnostic Explanation (LIME), to explain the behaviour of the supervised model, which essentially serves as a proxy for the clustering algorithm. To demonstrate the feasibility and effectiveness of their approach, the authors apply their methodology to the problem of QoE analysis for video quality under encrypted traffic scenarios. Their results demonstrate that EXPLAIN-IT can intuitively differentiate clusters based on features that are associated with video resolution groupings and poor network performance.

In [102], the authors develop a DNN classifier to predict the QoE perceived by EUs within a cellular network. However, as the opaque nature of the model does not allow them to see why certain EUs experience poor QoE, the authors propose the use of LIME [59], a prominent explanation method, to identify the factors of the input space that lead to poor QoE. In [103], the authors develop a tree-based model for predicting cellular coverage for an LTE network. By using the SHAP explanation method [66], the authors are able to determine which input features are most influential to the model's predictions (e.g., user distance to cell, cell ID and spatial characteristics) as well as to verify intuitive behaviours of the model, such as higher distances between the EU and serving cell resulting in monotonically decreasing Reference Signal Received Power (RSRP).

The general problem of traffic prediction has also seen interest from researchers aiming to enhance the explainability and interpretability of their solutions. For example, in [104], the authors revise some of the most current and up-to-date tools that can be used to foster greater robustness and explainability in 6G networks. As a case study, the authors apply the SHAP and LRP XAI methodologies to a network traffic forecasting problem and analyse the resulting explanations from a time series perspective. Their results reveal that both these methodologies are able to identify similar trends with respect to how recent and old data samples influence the model's behaviour. Moreover, the authors

demonstrate how the Gramian Angular Field (GAF) encoding method can be used to cross-validate the findings of both XAI methodologies.

2.2.1 Resource Reservation within Sliced Networks

In this subsection, we briefly review the topic of resource reservation within sliced networks, which later serves as the first use-case scenario explored within our work. Specifically, we begin this subsection by briefly reviewing the general concept of network slicing, and in particular, highlighting some of the many AI/ML solutions that have previously been proposed on this topic to date. Following this, we discuss any related research on XAI for resource reservation within sliced networks and outline the gaps that this research aims to fill in the later sections of this thesis.

Overview

In future cellular networks, Mobile Network Operators (MNOs) will be able to dynamically allocate dedicated resources to third-party service providers through the network slicing paradigm. Under this paradigm, MNOs who deploy their own physical network infrastructures and hold spectrum licenses, will be able to lease portions of their network resources, referred to as slices, to third-party tenants, such as vertical industries or Over-The-Top (OTT) service providers. These tenants can then use these resources to deliver dedicated network services to their end customers. At the heart of its success, network slicing relies on the MNOs' ability to share its limited resources among the multiple tenants in the network in a flexible and efficient manner. Additionally, it is expected that the network slicing framework will comprise both a forward market, where tenants may lease network resources from the MNO well in advance of the time at which these resources will be used, as well as a real-time market, where tenants can request additional resources from the MNO within a short time. Importantly, while the target of high flexibility can be achieved using technologies, such as Software-Defined Networking (SDN) and Network Function Virtualisation (NFV) to create logical networks, or slices, that can be dynamically tailored towards the requirements of each tenant, doing so efficiently requires additional tools which can accurately model future resource demands when and where they are needed by each slice tenant [105].

The dynamic and complex nature of this problem has spurred numerous AI/ML-driven solutions in the literature which aim to make use of the data-driven and model-free nature of these tools to aid in the resource allocation task. In general, the assumptions adopted by these frameworks can be classified according to two main approaches [106]. In the first case, the MNO is assumed to act as a centralised entity amongst the tenants [107]–[110]. By receiving only the requested KPIs from each tenant, the MNO is responsible for the complete E2E orchestration of the network resources and how they should be allocated amongst each tenant's end users in order to meet their specified KPIs. On the other hand, the second case assumes that the tenants themselves are responsible for the allocation of resources amongst their end users and for determining how many resources need to be reserved from the MNO to meet future demands [105], [111]–[114]. Perhaps unsurprisingly, each of these frameworks offers its own advantages and disadvantages to the resource allocation problem. For example, by allowing the MNO to act as the centralised entity, frameworks adopting the former case have a more globalised view on the network usage statistics and therefore, tend to achieve higher levels of resource utilisation and balanced scheduling between the tenants. In contrast, frameworks based on the second case take on the perspective of a single tenant, who only knows the usage statistics about its own end users whilst lacking any additional information regarding the demands of the other tenants within the network. Despite this apparent drawback, the authors of [112] have previously demonstrated that models from this framework can still yield satisfactory orchestration policies that have lower levels of Service-Level Agreement (SLA) violations due to the availability of rich application-layer information, which is typically not available to the MNO due to privacy-related legal barriers. Moreover, while works based on the first case represent a relatively mature area of research, fewer works have so far explored the benefits that can be gained in the second case, particularly within the context of network slicing. In Section 3, we consider the concept of network slicing from the latter perspective, which serves as the first use case explored in this work.

Related works, gaps & challenges

The survey paper by [111] has previously discussed some of the many studies that have employed AI/ML approaches to tackle resource allocation in sliced networks. In [105] the authors propose the use of a 3-dimensional CNN encoder-decoder model to predict

the future capacity demands of cellular network slices. In this framework, the CNN encoder is used to learn low-level features that have strong correlations between slice combinations, prediction horizons, and data centres. In addition to this, their decoder stage, which is responsible for predicting future demands, includes the use of a dedicated loss function that provides the MNO with the ability to adjust the trade-off between over-provisioning and under-provisioning (i.e. demand violations): unlike conventional regression tasks, where the objective functions tend to be based on symmetrical loss functions, such as Mean-Square Error (MSE), the trade-off between over and under-provisioning constitutes greater significance within slicing contexts as demand violations resulting from under-provisioning tend to result in larger fines to the MNO than in the case of revenue losses due to over-provisioned resources [115]. In [113], the authors propose a multi-stage framework for managing and allocating short-term resources to network slices. In this framework, a Gated Recurrent Unit (GRU) neural network is used to perform traffic prediction of the network, and subsequently used as input to an adjustment strategy stage, which is responsible for allocating resources amongst the various slices of the network. In addition to this, their ML pipeline also includes additional stages such as a cyclic training stage, which periodically retrains the GRU model to maintain high performance over time, as well as the use of uncertainty estimates alongside their point-predictions to aid in the allocation stage. In [114], the authors proposed a Deep Reinforcement Learning (DRL) algorithm used by the MNO for reserving fair amounts of resources for each of its tenants. In this framework, the MNO dynamically predicts the future traffic demands of each of its tenants in order to identify which tenants have over/under-provisioned their resources. Any over-provisioned resources are then collected back from the respective tenants and shared amongst the under-provisioned tenants using an allocation scheme based on the ratios of their minimum SLA requirements.

Despite the many papers mentioned above, the aspect of resource reservation for a network slice remains largely unexplored, with the majority of related work focusing on resource allocation from the perspective of the MNO or other central entities, such as capacity brokers [116]–[118]. In other studies which do consider the slicing concept, the main focus has been on the business and economic aspects of network slicing, for

example, in the form of revenue or cost optimisation for the MNO [119], profit maximisation for the tenants [120], or through the optimal design of the auctioning process [121], [122]. While the work of [123] has previously considered the reservation problem from a slice tenant's perspective, its scope is limited in the sense that it only considers reservations from a short-term perspective and also does not account for diversification in terms of QoS or cyclo-stationary availability of resources, both of which are highly relevant aspects of resource provisioning within the context of network slicing. Moreover, we note that the solution presented by [123] has been derived based on a heuristic approach that combines a closed-form expression with the Stochastic Gradient Descent (SGD) algorithm to determine its reservation policy for the tenant. In contrast, our work later explores the idea of employing deep learning solutions to develop a two-stage leasing scheme whereby the slice tenant can perform both long-term in advance reservations at a cheaper price as well as short-term on-demand requests at a higher price.

In addition to the many studies that have investigated the problem of Resource Management (RM) within sliced networks from a strictly AI/ML perspective, overall, little attention has been given to the inclusion of XAI within this main area. Moreover, we find that for the few works that have considered the use of XAI, none have adequately assessed the faithfulness of the underlying explanations which they present against a ground truth. For example, in [87], the authors outline the need for explainability in future 6G networks and propose a number of KPI metrics to evaluate the perceived level of trust associated with a particular model. Specifically, the authors propose the use of a trust broker entity to provide explanations to different stakeholders of the network and offer a number of KPI metrics to evaluate the level of trust associated with a particular model, as perceived from the HITL's perspective. The authors also provide a brief overview of how tools like LIME can be used to provide explanations to handover and resource allocation problems. However, their work does not include any experimental results for any of these scenarios nor do their proposed metrics evaluate the true faithfulness explanations in a purely quantitative manner. In recent work by [124], the authors compare the use of different explanation methods to perform root-cause analysis on a SLA violation prediction model. The results of their analysis found that SHAP provides the most consistent explanations when cross-referenced against a causal analysis tool. However, beyond a qualitative assessment of the explanations, their analysis does not

consider any quantitative metrics to assess the various methods investigated. In [125], the authors present a high-level framework for providing explainability to a Unmanned Aerial Vehicle (UAV) serving the role of a small cell, which must decide when to relocate to a different service region or to return to its base to recharge. At each service region, the UAV performs power allocation over a large number of ODFM channels. To achieve real-time optimization, the authors propose the use of a DNN to approximate the classic iterative water-filling power allocation solution, while a double duelling deep Q-network (DDDQN) is used to approximate the Markov decision process (MDP) for the UAV's flight actions. To enhance the explainability of the UAV's actions, the authors propose to employ local feature analysis to extract data features from the hidden layers of the neural network used to predict the Q-values of the agent's future states. Without explicitly discussing how explainability has been achieved in their work, the authors demonstrate how feature importance can be used to gain insights into the UAV's decision-making process. For example, they find that factors such as higher unmet load, high battery power and high QoS complaints may cause the UAV to abandon its current position in order to find a location where better service can be delivered.

Overall, little attention has been given to the use of XAI in network RM problems in sliced networks. Moreover, of the works discussed above and throughout this entire Section, none assesses the quality or faithfulness of the explanations in a quantitative manner. That is, in all the cases discussed so far, the explanations are either considered to be trustworthy to the models they are explaining or qualitatively verified by cross-referencing against other methods.

2.2.2 Network Intrusion Detection & Cybersecurity: Overview, Gaps & Challenges

In this subsection, we briefly review the topic of network intrusion detection within the cybersecurity space, which later serves as the second use-case scenario explored within our work. Specifically, we begin this subsection by briefly reviewing the general concept of network intrusion detection, and in particular, the role that AI/ML has played in this established topic of research. Following this, we then discuss any related research on XAI for network intrusion detection and outline the gaps that this research aims to fill in the later sections of this thesis.

Overview

In recent years, there has been an alarming increase in the number of cybersecurity threats witnessed around the globe, with each successful threat posing severe risks to the financial and reputational stability of the companies and organisations affected. In a general sense, cybersecurity deals with the safeguarding of any devices, software, or data within a network from being maliciously accessed and exploited by intruders [126]. As an established topic of research spanning well over three decades, the field of cybersecurity remains at the top of mind for both the academic community and the entire ICT sector, who face a constant battle to develop better and more up-to-date solutions to combat the on-going efforts of attackers. This constant need for improving our cybersecurity infrastructure has been reaffirmed by the findings of many recent threat reports, including IBM's Cost of Data Breaches report [127], which has estimated the average cost of data breaches in 2022 to have climbed to almost 4.35 million USD, while further findings by Mandiant [128], have revealed an almost doubling in the number of zero-day exploits occurring in 2021 compared to 2019, to give just a few examples.

As a key defence against cybercriminals, NIDSs comprise any hardware or software-based algorithm that actively or passively analyses the flow of traffic across a network to determine whether a potential attack is taking place [129]. In the past, various classical ML techniques, such as Naïve Bayes [130], Random Forest [131], Decision Tree [132], SVM [133], and MLP [134], have been employed to detect malicious flows. However, faced with exponential increases in traffic volume, coupled with a continued emergence of new attack strategies from intruders attempting to evade detection, cybersecurity experts have gradually begun shifting towards the use of deep learning techniques, such as DNNs [135], and ensemble learning methods, such as XGBoost [136], in an effort to maintain adequate levels of accuracy.

In the cybersecurity literature, two main approaches dominate the design of a NIDS [137]. In the first approach, the NIDS is designed in a supervised manner to detect whenever a traffic flow exhibits signatures which are similar to those appearing alongside *known* attacks. In the second approach, unsupervised ML methods are used to learn statistical or latent representations of normal traffic, and anomaly detection methods are subsequently employed to detect whenever a flow differs significantly from this baseline.

Importantly, while NIDSs based on the former approach have shown general success against known attacks, they often fail when presented with zero-day attacks [126]. On the other hand, recent works adopting deep learning solutions for anomaly-based NIDS, such as deep autoencoders [138]–[140], have demonstrated promising results against detecting new attacks.

Related works, gaps & challenges

In addition to high detection accuracy, many cybersecurity experts now also consider notions of explainability and interpretability to be an essential characteristic of a robust NIDS. This is particularly the case for NIDSs based on deep learning models, as their inherent 'black-box' nature means that even once malicious traffic has been successfully detected, considerable human effort is still required to determine *why* the flow is malicious, and therefore, how to best deal with the attack [141]. Combined with the recent surge of XAI techniques appearing across the wider scientific domain, a new wave of cybersecurity works has appeared in recent years that offer additional layers of explainability and interpretability for the HITL. The survey papers by [88], [142], [143] provide a thorough overview of the state-of-the-art within this fast-expanding landscape. In general, the vast majority of literature on this topic tend to focus on the application of XAI to directly inform the HITL about the model's decision-making process and to enhance the overall user trust in the proposed solution. For example, in [144], the authors have presented one of the earliest frameworks based on the SHAP methodology for cybersecurity. By applying their framework to two separate models (i.e., a one-versus-all NIDS and multiclass NIDS), they demonstrate how local and global SHAP explanations can help guide cybersecurity analysts on how best to improve the structure of their models. In [145], the authors have recently presented a framework called "FAIXID", which aims to assist analysts of various roles and expertise to make more efficient and informed decisions when attempting to isolate credible attacks on the network from false positives output by their intrusion model. Their proposed framework combines data mining techniques alongside XAI methodologies to evaluate and select the best set of explanations to suit the particular expertise of the HITL. As another example, the authors in [146] have recently proposed a framework for explainable anomaly detection based on the LRP methodology [57]. By manually analyzing the outputs of each explanation, they are

able to identify and verify the key features used by their model in classifying distributed denial-of-service (DDoS) attacks, such as the amount of data transferred, the number of connections made, the connection frequency, etc.

While many works, such as those mentioned above, tend to utilize explanations to offer direct insights to the HITL, a few additional studies leverage explanations to explicitly improve the performance of their final model. As one of the earliest examples that we are aware of, the authors in [147] have previously presented a strategy in which they aim to use explanations to perform targeted improvements of their model. Specifically, their proposed strategy consists of i) isolating all samples of the training set that have been misclassified by the model, ii) identifying the most important features of these samples, i.e., with the importance indicated by the corresponding explanation, and iii) by arguing that these features have essentially 'distracted' the model from focusing on the truly relevant features needed to make a correct prediction, the authors then propose to create a new dataset where noise has been added to the most important features identified in step ii, before iv) re-training the model for a limited number of epochs so that the model learns not to rely on the same misleading features in the future. Based on this strategy, the authors show that it is possible to greatly reduce the number of false positives and false negatives produced by their model. We categorise this general approach taken by [147] as being closely related to a model re-training scenario. Under a similar categorisation, we also mention the recent works of [148] and [149], in which the authors of both studies propose the idea of manually analysing explanations to perform feature selection and ultimately improve the accuracy of their fine-tuned model.

So far, our literature review into the application of XAI for network intrusion detection suggests that little to no work has previously examined the potential benefits that can be gained when additional AI/ML tools are leveraged to process the explanations automatically. To the best of our knowledge, only the previous work of [150] has looked at this potential gap. In this study, the authors aim to minimize the effort required by a human analyst to manually investigate each sample that has been deemed malicious by a primary anomaly detection model. To this end, the authors propose the use of a second ML model which is trained in a supervised manner using labelled input samples and their corresponding explanations from the primary model, to detect wrongful classifications from the primary model. However, we note that the results presented in [150]

offer only a limited glimpse into the potential benefits of leveraging XAI coupled with additional ML/AI to produce performance gains. In addition, the authors also propose the use of supervised ML to process the information within the explanations. Fundamentally, this requires a HITL to manually label each of the samples used for training, while the use of a supervised approach, i.e., over an unsupervised one, may potentially lead to poor performance in detecting attacks that lie outside of the model's training distribution [126].[4]

2.2.3 Section Summary

The growing need for transparency and understanding in how complex models and algorithms make decisions has led to a diverse range of explainable and interpretable AI solutions proposed across various sectors of the telecommunication domain. Table 2.1 categorises the explainable and interpretable AI papers discussed in this section based on their primary application areas. Additionally, Table 2.2 further classifies each of these papers according to the main XAI tools they employ, following the general XAI taxonomy previously introduced in the first section of this chapter. Overall, the black-box problem has received significant attention across multiple telecommunications applications, including cybersecurity, network service quality, signal interference management, resource management, power allocation, and end-to-end optimization at the physical layer. Moreover, various explainable and interpretable AI techniques have been explored within each of these domains, with local and global feature importance emerging as the most widely adopted XAI approaches. Meanwhile, shallow models, physics-inspired ML, and analytical expressions have all been considered for providing interpretable AI solutions.

Despite the critical role of AI-driven automation in resource management and its potential impact on network reliability, our review highlights a concerning lack of XAI solutions in this sector. Among the limited studies applying XAI concepts, feature importance has been the dominant technique with methodologies such as LIME being proposed for handover and resource allocation [87], SHAP employed for explaining SLA violation predictions [124] and the general concept of feature importance proposed for explaining the actions of a Reinforcement Learning (RL) agent operating as a UAV-based small cell

[4]We note that this work was published concurrent to our own work presented in Chapter 4.

[125]. However, none of these studies have rigorously validated the faithfulness of their explanations against ground truth, posing potential risks towards the adoption of AI solutions for automated network management by fostering a false sense of security in their decision-making processes.

On the other hand, the cybersecurity domain has received a tremendous surge of interest from the XAI community. However, most studies in this field focus primarily on using XAI to inform human decision-makers and build user trust (e.g., [144]–[146]). Only a limited number of studies have investigated the idea of leveraging explanations to explicitly improve the performance of their final model, with all requiring a substantial amount of manual effort to assist in tasks of feature selection and model re-training (e.g., [147]–[149]). To the best of our knowledge, only the work of [150] has previously examined the potential benefits that can be gained when additional AI/ML tools are leveraged to process the explanations. However, their study provides limited results and relies on supervised ML to process the explanations, which also requires extensive manual labelling.

In the following chapters, we address these gaps by proposing an explainable AI-based framework for network slicing, incorporating explanation quality metrics with ground truth (Chapter 3). Additionally, we introduce an explainable AI-driven methodology leveraging unsupervised ML to automatically enhance network intrusion detection systems (Chapters 4 and 5).

TABLE 2.1: Classification of related work based on application area.

Problem Area	**Paper**
Physical Layer (E2E)	[89], [90], [91]
Power Allocation	[93], [46], [92]
Signal Interference	[96], [97], [95], [98], [99]
Network Service Quality	[101], [102], [103], [100], [104]
Resource Management	[87], [124], [125]
Cybersecurity	[141], [144], [145], [146], [147], [148], [149], [150]

TABLE 2.2: Classification of related work based on XAI taxonomy.

Paper	**Post-hoc**	**Ante-hoc**	**Local XAI**	**Global XAI**	**Summary of Approach**
[89], [90], [93], [95]		✓			Shallow Models
[91], [97], [100], [141]	✓			✓	Statistical
[92], [99]		✓			Physics-Inspired
[46], [98]		✓			Analytical
[96], [125], [147], [150]	✓		✓		Feature Importance
[87], [101]–[104], [124], [144]–[146], [148], [149]	✓		✓	✓	

Chapter 3

Resource Reservation within Sliced Networks

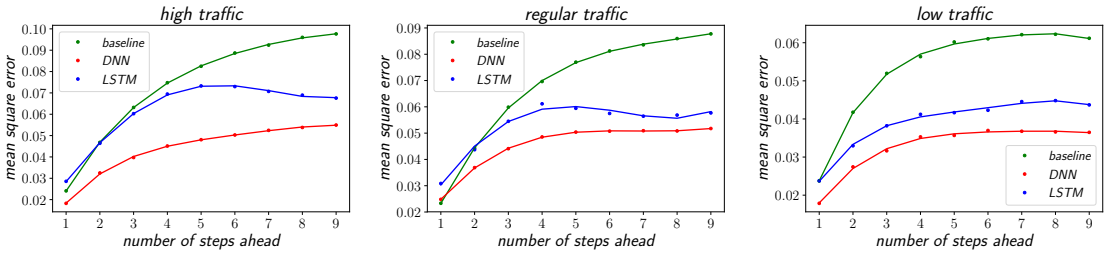
3.1 Introduction

In this chapter, we present our first contribution, C1, which partially addresses RQ1 and directly addresses RQ2. In developing our contribution, we consider the problem of resource reservation from the perspective of the slice tenant. In this scenario, we envision a cellular network in which the MNO shares its network resources, i.e., in the form of bandwidth capacity, simultaneously among its multiple slice tenants. At discrete time intervals, each tenant competes for the availability of the MNO's future resources by carefully placing reservation requests that aim to fulfil the future traffic demands of the tenant's own end users. Moreover, each tenant has access to two types of resources representing portions of the MNO's overall capacity: guaranteed resources and unreserved best-effort resources. In this sense, guaranteed resources are those that the tenant has previously booked in advance to be made available at the current time window, while best-effort resources represent portions of the spare capacity (after pre-reserved resources have been accounted for) that the MNO shares between each of its tenants at the start of each time window. Importantly, we assume that guaranteed resources always come at a higher cost than best-effort resources, and therefore, to minimise its costs, the slice tenant is inclined to reserve guaranteed resources only when it expects that best-effort resources will not suffice to meet its own end user's demands.[1]

[1] In Section 3.3, we provide a more formal and in-depth description of the network slicing system model used throughout this section.

Under this network slicing scenario, an initial study was conducted to investigate the feasibility of employing ML solutions to aid the tenant in its task of reserving resources from the MNO [151]. Specifically, two deep learning solutions comprising both a vanilla DNN and LSTM architecture were chosen to investigate the potential benefits that ML can have within the network slicing context in which each tenant is responsible for reserving its own future resources from the MNO. For each of these solutions, a dataset of real-world cellular traffic data was employed to optimise each model to perform the task of multi-step ahead resource reservation under three main network traffic conditions, including high traffic, regular traffic, and low traffic demand profiles.

Figure 3.1 presents a subset of the overall findings that were obtained following this initial study.[2] In this figure, the overall performance of both ML solutions have been evaluated based on the MSE arising between the reserved resources of each ML model in a multiple-step ahead scenario and the actual amount of resources that were subsequently required to meet the demands of the slice tenant. Additionally, the performance of both models are compared against that of an ARIMA baseline modelled under ideal information. From these initial results, it is evident that both ML solutions are capable of significantly outperforming the ARIMA baseline, especially as the prediction horizon increases.



(A) High-traffic (B) Regular-traffic (C) Low-traffic

FIGURE 3.1: Comparing performance of different ML solutions for multi-step ahead resource reservation under different traffic conditions using Mean-Square Error (MSE).

While this initial work provides a clear motivation for the use of ML solutions for resource reservation to achieve greater accuracy, however, there still remains the challenge of understanding the underlying rationale of the reservation models. In the following sections of this chapter, we aim to address this shortcoming by proposing a methodology encompassing state-of-the-art XAI techniques to explain and enhance the transparency

[2]We refer the reader to [151] for full details on the training and evaluation aspects of this study

of black-box algorithms employed in complex resource management tasks, such as resource reservation. Specifically, we begin by outlining the general components of our XAI methodology in Section 3.2. Following this, we outline our system model for a sliced network in Section 3.3. Based on our system model, we then formulate a deep-learning solution for Short-Term Resource Reservation (STRR), which is a special type of reservation model where requests are made primarily in the near future, such as the next immediate time step, in Section 3.4. In Section 3.5, we evaluate the various components of our work, beginning with an evaluation of our STRR model's performance, followed by an in-depth demonstration of our XAI-based methodology, i.e., when applied to our STRR model as a use case example. Finally, in Section 3.6, we provide an in-depth complexity analysis of our evaluation process, which we employ in the previous subsection to assess the faithfulness of the explanations produced by our methodology. The main contributions of this chapter can be summarised as follows:

1. We outline a methodology based on state-of-the-art XAI techniques, including both local [54] and global explanation techniques [81], to explain the behaviour of AI models used in resource management problems for sliced networks, such as STRR.

2. We present a network slicing system model whereby a MNO leases its resources in the form of bandwidth capacity among a set of slice tenants, who then compete for these resources based on their own resource reservation policies.

3. We propose a reservation policy that a single tenant of interest may implement based on the available feedback it receives from the MNO at each time step or allocation window, such as the amount of best-effort, delivered/undelivered traffic, etc. Utilising this policy, we subsequently train an ML solution that our slice tenant can employ to perform STRR and further evaluate its performance across an open-source dataset of real-world traffic traces.

4. Using our reservation model as a use case example, we demonstrate how explanations proposed in our XAI methodology can be used to monitor the real-time decisions of the model, to reveal global trends about the model, as well as aid in the diagnosis of potential flaws during its development.

5. We quantitatively validate the faithfulness of our explanations using an extensive range of XAI metrics [81], [152].

While the idea of providing XAI to the network slicing domain has been explored in a limited number of previous works (we refer the reader to Section 2.2.1 for more details), to the best of our knowledge, point 5 represents a first in the use of quantitative XAI metrics with a proposed ground truth to assess the faithfulness of explanations produced in the context of network slicing, and possibly, also the general telecommunications field.

3.2 XAI Methodology

In this subsection, we outline the details of our proposed methodology to provide explainability to complex models employed in network resource management tasks, such as resource reservation. During the initial development of this methodology, we have considered various techniques from the field of XAI, and in particular, how these techniques can be combined to provide explanations which are both intuitive to the HITL to understand and interpret as well as techniques which are capable of producing explanations that are faithful to the models they represent. In addressing these criteria, we focus the scope of our work on the prominent class of AFA XAI techniques, in which explanations take the form of feature "importance scores" indicating the contribution each feature has towards a particular output of the model. In Fig. 3.2, we summarise the main components of our methodology, which essentially forms an extension of the conventional pipeline for AI/ML by proposing three additional XAI-related stages, including an initial stage that computes local explanations of the model's outputs, a second stage that combines local explanations to produce global explanations of the model's general behaviour, and a final stage which quantitatively assesses the faithfulness of the resulting explanations to ensure that they remain trustworthy and actionable in a real-world context. In what follows, we elaborate further on each of these three stages:

3.2.1 Local Explanations

For the purpose of this work, we compute local explanations of our RM model by considering the Kernel SHAP method from [54]. We consider this method primarily as it offers a strong theoretically-grounded framework for computing accurate local explanations of

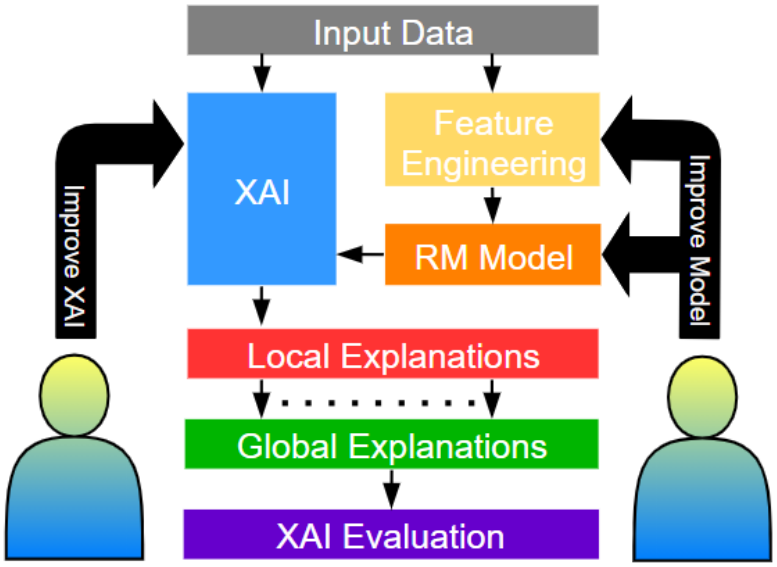


FIGURE 3.2: Summary of our proposed methodology.

any black-box model (for more details, we refer the reader to Section 2.1.4). However, in general, any XAI method adhering to the class of local AFA explanations can be substituted into the proposed methodology.

When considering AI/ML models used in resource management, the Kernel SHAP method can be used to understand the *magnitude* and *direction* by which each feature impacts a particular decision of the model. For example, if we consider a simplified model for STRR, where the inputs to the model may include a tenant's past traffic demand and undelivered traffic, and where output corresponds to the amount of resources the tenant should reserve in the near future, i.e., in units of Mb, then a local explanation could, for instance, be used to indicate that knowledge of the past traffic demand has caused the model to increase its reservation by +12 Mb, while knowing the undelivered traffic has reduced it by -8 Mb. As we later demonstrate in Section 3.5, the ability of local explanations, such as Kernel SHAP, to essentially *decompose* the output of the model among the individual influences caused by each input feature makes the use of local explanations especially suited towards assisting the HITL to understand and validate the behaviour of the model against their own mental model.

Throughout this section, we refer to the local importance scores calculated by the Kernel SHAP method simply as "SHAP" values. Moreover, we recall from Section 2.1.4, that in its general form, when we consider a black-box model that takes as input an N-dimensional vector of features, $\mathbf{x} = [x_1, ..., x_N]$, and produces a single output, $y = f(\mathbf{x})$,

then a local AFA based explanation, such as SHAP, comprises of a vector of SHAP values, $[\phi_1, ..., \phi_N]$, where each x_i is associated with its own particular ϕ_i. Moreover, in the case of Kernel SHAP, each ϕ_i associated with feature x_i can be interpreted as the change in the expected output of the model when x_i is known to the model versus when x_i is unknown to the model. In addition, the "local accuracy" property of the SHAP values ensures that the sum of all SHAP values for a specific instance adds up exactly to the difference between the model's output for that instance and the expected output of the model, i.e.,

$$\sum_{i=1}^{N} \phi_i = y - E[f(\mathbf{x})]. \tag{3.1}$$

3.2.2 Global Explanations

To generate global explanations of our model, we consider the use of SHAP dependence plots and SHAP summary plots in our work. Essentially, a SHAP dependence plot can be used to understand the marginal effect that a specific feature has on the output of the model. For a given, feature, a SHAP dependence plot is constructed by plotting its local SHAP values (vertical axis) against the values that the feature takes (horizontal axis) across multiple instances of the dataset. In the case of a SHAP summary plot, a series of subplots are used to indicate how the SHAP values of each feature are distributed according to their magnitude, density and direction [81].

In the context of resource management, such as STRR, global explanation plots can aid the HITL to verify that the trained model correctly learns relationships experienced in the physical world, such as an STRR model that monotonically increases its reservation as the past traffic demand increases. In addition, global explanations can also aid in revealing potential bugs which may be harmful to the model's performance: in Section 3.5, we demonstrate an example of how SHAP dependence plots allowed us to debug flaws during the early design stages of our STRR model. We note that, while the use of XAI for model debugging has seen growing interest from within the general XAI community in recent years [153], its application in the telecommunications domain remains largely unexplored to date.

3.2.3 XAI Evaluation

As the final stage in our methodology, we propose to validate the faithfulness of the explanations by considering an extensive range of XAI metrics. While many forms of qualitative evaluation strategies have previously been employed in related literature (we refer the reader to the reviews in Sections 2.1.5 and 2.2.1), in this work, we are particularly interested in quantitative metrics which are able to compute the faithfulness of the explanations produced by our methodology relative to an absolute scale or ground-truth score. To the best of our knowledge, no previous set of XAI metrics have been proposed which can compute the faithfulness of local explanations against a ground-truth score. Instead, we propose to solve this shortcoming by adopting a state-of-the-art set of evaluation metrics and proposing our own method for computing the corresponding ground truth of each metric. Specifically, we propose the use of the XAI metrics previously presented in [81] as a starting point for achieving the goals of our evaluation stage (we refer the reader to Section 2.1.5 for more details on these metrics).

As elaborated in [81], the majority of these metrics operate on a common basis, whereby the most important features (as indicated by the corresponding local explanation) are 'removed' from the model, i.e., by mean-masking features with the highest or lowest corresponding SHAP values, and subsequently observing the impact that this has on the output of the model. Essentially, in their originally presented form in [81], these metrics can solely be used to cross-validate the relative performance of different XAI methods against one another. Apart from only one of the presented metrics in [81] (i.e., the 'Local Accuracy' metric), all of the remaining metrics lack an absolute scale or ground truth in their calculation, which becomes essential when the true faithfulness of the explanations have to be correctly assessed. In our work, we propose to compute this ground truth by exhaustively permuting across all possible combinations of feature importance rankings and sign, which form the basis of how all of these metrics operate.[3] The final result allows us to compare how well each explanation performs compared to the absolute score, and therefore, whether or not the explanation should be trusted and acted upon by the HITL. In Sections 3.5.4 and 3.6, we provide a detailed overview of how we implement this stage in our work as well as an analysis of the associated computational complexity

[3] We note that this process requires complexity of the order $\mathcal{O}(NN!)$.

which it calls for, respectively

In the next subsection, we outline the steps of our system model for resource reservation within a sliced network environment.

3.3 System Model for Resource Reservation in Sliced Networks

We consider a sliced network comprising a single MNO serving the traffic demands of a set of tenants across its network infrastructure, where we use the term $k \in K$ to denote the k^{th} tenant from the full set of tenants, K, served by the MNO. Here, we outline our system model according to a single base station, but note that the presented solution can be replicated across multiple base stations. We also assume that the MNO allocates its network resources within discrete scheduling windows or time slots. Below, we outline four main tasks that the MNO performs during an arbitrary scheduling window occurring at time t:

(a) *Schedules Future Resources:* The MNO receives reservation requests from each tenant, where $r_{t+i,k}$ denotes the amount of resources, expressed in terms of bandwidth capacity measured in units of megabytes (Mb) per time slot, which tenant k wishes to reserve during future time slot $t+i$. Provided the total sum of requested resources from all tenants for time $t+i$ does not exceed the MNO's maximum capacity, c, the MNO schedules the requested resources of each tenant for future use. If the sum of requested resources exceeds the maximum capacity, the MNO distributes its resources proportionally amongst the tenants such that the actual amount of resources scheduled for tenant k at future time $t+i$, is given by:

$$a_{t+i,k} = \begin{cases} r_{t+i,k}, & \sum_{k \in K} r_{t+i,k} \leq c \\ \dfrac{c \cdot r_{t+i,k}}{\sum_{k \in K} r_{t+i,k}}, & \text{else} \end{cases} \tag{3.2}$$

(b) *Distributes Best-Effort Traffic:* The MNO distributes any spare capacity it has among the under-provisioned tenants in a best-effort manner. Using $v_t = c - \sum_{k \in K} a_{t,k}$ to denote the MNO's spare capacity at time t, $s_{t,k}$ to represent the actual traffic demands of tenant k at time t, and $h_{t,k} = max(s_{t,k} - a_{t,k}, 0)$ to signify the amount

by which tenant k is under-provisioned at time t, the best-effort traffic assigned to tenant k is given by:

$$b_{t,k} = \begin{cases} 0, & h_{t,k} = 0 \\ min\left(h_{t,k}, \frac{v_t}{|K|}\right), & \text{else} \end{cases} \tag{3.3}$$

(c) *Charges Each Tenant:* Tenants are charged according to their future reserved resources and those which have been assigned to them as best-effort. Although specific pricing strategies are beyond the scope of this work, we assume that the unit cost of reserved resources is much higher than for those allocated in a best-effort manner. Thus, in developing its own reservation model, a tenant should aim to minimise the price it pays for resources, while maximising the QoS to its end customers by reserving only the amount of resources that are in excess of what it expects to be available in a best-effort manner.

(d) *Provides Feedback:* At the end of the scheduling window, the MNO provides feedback information to each tenant regarding the network statistics of its own slice. This information includes their total traffic demand $s_{t,k}$, best-effort traffic $b_{t,k}$, undelivered traffic $u_{t,k}$, delivered traffic $d_{t,k}$ as well as the number end users that are active in their slice $z_{t,k}$. Using equations (3.2) and (3.3), the delivered and undelivered traffic of tenant k can be derived as:

$$d_{t,k} = min(a_{t,k} + b_{t,k}, s_{t,k}) \tag{3.4}$$

and

$$u_{t,k} = s_{t,k} - d_{t,k}. \tag{3.5}$$

We summarise the main parameters of our system model for network slicing in Table 3.1 below:

TABLE 3.1: Summary of System Variables

Parameter	Description
k	k^{th} tenant
$r_{t+i,k}$	Amount of resources tenant k wishes to reserve at time $t+i$
c	Maximum capacity of network
$a_{t+i,k}$	Amount of resources scheduled for tenant k at time $t+i$
v_t	Spare capacity at time t
$b_{t,k}$	Best-effort traffic of tenant k at time t
$s_{t,k}$	Total traffic demand of tenant k at time t
$u_{t,k}$	Undelivered traffic of tenant k at time t
$d_{t,k}$	Delivered traffic of tenant k at time t
$z_{t,k}$	Number of active end users in the k^{th} slice at time t

3.4 A Deep Learning Solution for STRR

In this subsection, we describe the implementation details of our STRR model. Specifically, we begin by first formulating the reservation task from the tenant's perspective and deriving an expression for the optimal ground-truth labels required to train our solution. We then outline the design considerations and parameters of our deep learning solution for STRR, and finally, we present an overview of the dataset used in our work.

3.4.1 Problem Formulation

We consider the reservation task, T, in which a slice tenant, k, reserves resources from the MNO to meet the future demands of its end users. Specifically, the tenant wishes to minimise its expenditure for resources (i.e., by avoiding over-reservations), while maximising the QoS to its end users (i.e., by avoiding under-reservations). We cast this problem into a supervised learning task, where the objective is to achieve a good trade-off between expenditure for resources and QoS. Using the aggregated traffic traces of each tenant and system model from Section 3.3, we construct the ground-truth labels of our model by simulating the behaviour of an *oracle*, which has full knowledge of the network's parameters and future states, and makes informed reservations for our tenant k, given by:

$$r^*_{t,k} = \begin{cases} 0, & \frac{c - l_t}{|K|} \geq s_{t,k} \\ min\left(s_{t,k}, \frac{|K|s_{t,k} + l_t - c}{|K|-1}\right), & \text{else} \end{cases} \tag{3.6}$$

where for brevity, we use l_t to denote the total sum of reservations from all *other tenants* excluding our own at time step t, i.e.,

$$l_t = \sum_{i \in K \setminus \{k\}} r_{t,i}. \tag{3.7}$$

Intuitively, this policy urges our tenant not to make reservations when best-effort resources are large enough to cover its future traffic demand. When the availability of best-effort resources is limited, i.e., due to heavy competition in the network, the model will aim to reserve at most the tenant's own future demand, or else the minimum portion of it that cannot be covered by the resources available on a best-effort basis. The outcome of this simulation produces the ground-truth labels and corresponding signals $s_{t,k}$, $u_{t,k}$, $b_{t,k}$, $d_{t,k}$, and $z_{t,k}$, which our tenant receives when making reservations under this policy. We note that for the purpose of this simulation, we assume that the other remaining tenants are perfect predictors of their own traffic, and thus make reservations according to $r_{t,i} = s_{t,i}, i \in K \setminus \{k\}$. We also set the maximum capacity of the network equal to the 80th percentile of the total traffic occurring during the training data, i.e., $c = P_{80}(\sum_{i \in K} s_{t,i})$, which ensures that the training set includes an adequate number of samples where the demand has surpassed the total maximum capacity of the network.

3.4.2 Model Design

For the design of our model, we implement a fully connected DNN with 3 hidden layers between the input and output layer. As inputs to the model, we consider six sources of information available to our tenant at time t, including its most recent traffic demand, best-effort traffic, undelivered traffic, number of end users, as well as two time-based features capturing seasonal trends at a daily and weekly level by considering the number of minutes passed since the start of each day (i.e. 00:00) and start of each week (i.e., Monday), respectively.[4] We also standardise each feature to have zero mean and unit variance before being passed as inputs to the model. Finally, the output of the model consists of a single scalar representing the reservation for the next scheduling window at time $t + 1$.

[4]We do not include the tenant's delivered traffic as a feature, as this becomes redundant to the model once the total and undelivered amounts are given.

To train our model, we use the TensorFlow library for Python with Keras backend [154]. We optimise the parameters of the model using SGD with a Huber loss function. This loss function is characterised by a 'delta' parameter which specifies a threshold at which the loss changes from a quadratic to a linear scale; as we find that the traffic data exhibits rare peaks that harm the model's convergence, the Huber loss presents a better alternative to the conventional L2 loss, which is known to be sensitive to outliers in the data. To determine a suitable value for this parameter, along with other hyper-parameters of the model, such as the number of neurons per layer, activation function (sigmoid, ReLU or tanh), and dropout rate for each layer, we perform a grid search of the model space using the Bayesian optimiser from the Keras Tuner library [155]. In Table 3.2, we summarise the grid search and final parameters of our model. Our final model consists of 5 layers, including the input layer, 3 hidden layers of sizes 16, 256 and 112 neurons, respectively, and an output layer. For each of the hidden layers, we use a ReLU activation function, as well as apply drop out of strength 0.1, 0.5 and 0.1, respectively. We train our final model for 1000 epochs using an early stop criteria of 50 epochs on the validation loss, and use mini-batches of size of 288 samples in order to improve the convergence rate during training.

TABLE 3.2: Model Summary

Parameter	Grid search space	Final
Input layer (Layer 1)	-	8
Dense (Layer 2)	16 - 256 (step = 16)	16
Activation (Layer 2)	sigmoid, ReLU, tanh	ReLU
Dropout (Layer 2)	0.1 - 0.5 (step = 0.1)	0.1
Dense (Layer 3)	16 - 256 (step = 16)	256
Activation (Layer 3)	sigmoid, ReLU, tanh	ReLU
Dropout (Layer 3)	0.1 - 0.5 (step = 0.1)	0.5
Dense (Layer 4)	16 - 128 (step = 16)	112
Activation (Layer 4)	sigmoid, ReLU, tanh	ReLU
Dropout (Layer 4)	0.1 - 0.5 (step = 0.1)	0.1
Output layer (Layer 5)	-	1
Delta	1, 10, 100, 250, 500	1.0

3.4.3 The Dataset

We use a publicly available dataset [156] of cellular traffic data recorded from over 5,000 mobile phone users during 2014. Each record contains application-layer data about the

end users' traffic, which can be mapped directly to their own service slice. To use this dataset in our work, we first aggregate each user's traffic trace into 5-minute intervals (i.e., the length of a scheduling window in our model) and combine the uplink and downlink traces of each application into a single trace. The resulting dataset gives the traffic demands subsequently used to compute the ground-truth labels using equation (3.6) as well as the number of end users that are active in each slice during each scheduling period. Finally, we use the 'Chrome' trace from this dataset as the tenant used by our model, and split the dataset into three sections, using approximately eight weeks of data for training (September 1st up to October 24th), one week for validation (October 25th up to October 31st), and one month for testing (November 1st up to November 30th). In Fig. 3.3, we illustrate a sample of the resulting traffic traces for 4 out of the 10 applications contained within the final dataset.

In the next subsection, we evaluate the performance of our STRR model and demonstrate the results of our XAI methodology applied to explain the STRR model's behaviour.

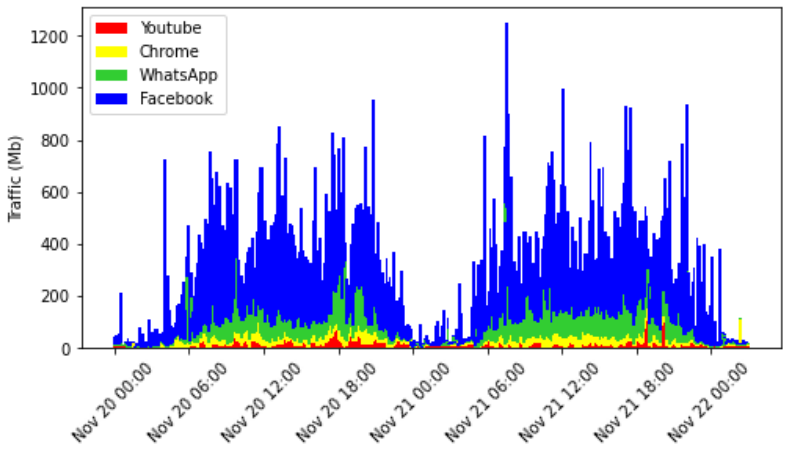


FIGURE 3.3: Example traffic traces from our dataset.

3.5 Results & Discussion

3.5.1 Performance of STRR model

We evaluate the performance of our STRR model using a series of KPI metrics, including RMSE, and over/under-reservation, as defined by equations (2.3), (2.4) and (2.5), respectively. For comparison, we also show the performance of a naive solution, which always reserves the most recent traffic demand for the next scheduling window. As shown in

Table 3.3, our model is able to outperform the naive solution in terms of RMSE and over-reservation, achieving a relative performance gain of approximately +14.7% and + 20% on these metrics, respectively. On the other hand, our model slightly underperforms in terms of under-reservation, achieving a relative gain of -11% in comparison to the naive solution.[5]

TABLE 3.3: Model Performance

Metric	Model (Mb)	Naive (Mb)	Rel. Performance Gain [[1-Model/Naive] · 100] (%)
RMSE	**16.3**	19.1	+14.7
Over-Reservation	**9.2**	11.5	+20.0
Under-Reservation	-13.1	**-11.8**	-11.0

3.5.2 Local Explanations

Fig. 3.4 illustrates an example of a local explanation of our STRR model. In this example, each bar corresponds to the SHAP value of one of the model's inputs, where red bars indicate features that raise the model's output above its expected value of ≈ 17.1 Mb, while blue bars indicate features that lower it. Specifically, the explanation for this sample indicates that the time of day/week, number of end users, and past traffic demand are responsible for increasing the model's output above its expected value, with the past traffic demand being responsible for the highest increase of ≈ 5.2 Mb. On the other hand, the best-effort and undelivered traffic are responsible for lowering the output of the model, with best-effort responsible for lowering the output the most by ≈ 5.9 Mb.

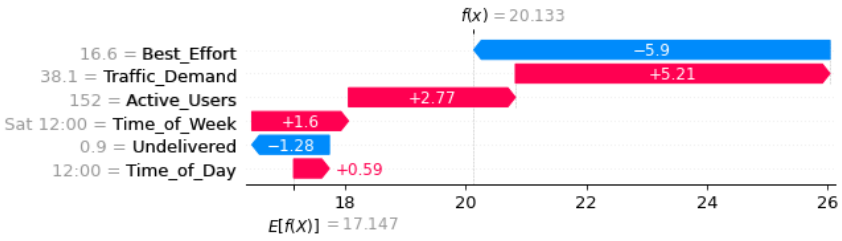


FIGURE 3.4: Example local explanation using Kernel SHAP.

[5]We consider the use of the naive predictor as a baseline as we are primarily interested in the explainability aspect of the STRR model, and not its sheer performance. We note that similar arguments have also previously been made for the use of the naive predictor as a baseline in related works on explainable traffic classification [157].

Importantly, in the context of resource management, local explanations can be used by the HITL to *monitor* and *verify* the real-time decisions of the model against their own intuition or mental model. For instance, in the case of our own STRR model, this process may involve considering the tenant's traffic demand of 38.1 Mb (as shown in Fig. 3.4), which in this case can be considered relatively high compared to the average traffic demand of $\approx$ 26.7 Mb. In this sense, as the explanation indicates that knowledge of the traffic demand has led to an increase in the model's reservation, this suggests that the model has interpreted high traffic demand to be associated with reduced network availability, which the model has appropriately compensated for by increasing its reservation so as to minimise any service loss. Similarly, by extending this analysis towards the remaining features of the model, it can be verified that the model appropriately increases its reservation based on the current time of day/week and relatively high number of end users, while the relatively high best-effort and low undelivered traffic reduce the reservation amount accordingly.[6]

3.5.3 Global Explanations

By combining local explanations from multiple samples, further insights can be gained into a model's general or global behaviour. In this work, we consider two global XAI methods applied to our STRR model, including a SHAP summary plot and SHAP dependence plots proposed in [81]:

In Fig. 3.5, we show a SHAP summary plot of our STRR model computed using samples taken from the test data. In this plot, each row or subplot corresponds to a distribution of SHAP values for one of the model's features, where dots forming each distribution correspond to local SHAP values of that feature computed across individual samples of the data. To produce the shape of each distribution, dots of similar magnitude (shown on the x-axis) are piled together to reveal dense regions, while the direction of the distribution is indicated by encoding each dot using a bi-variate colour scheme, with red dots denoting features whose values are above their expected amount and blue denoting those below it.

[6]We note that the analysis described here serves only as an example based on our own STRR model. In general, the manner by which one might verify the decisions of their own model will depend on the nature of the problem as well as the domain knowledge of the HITL performing the analysis.

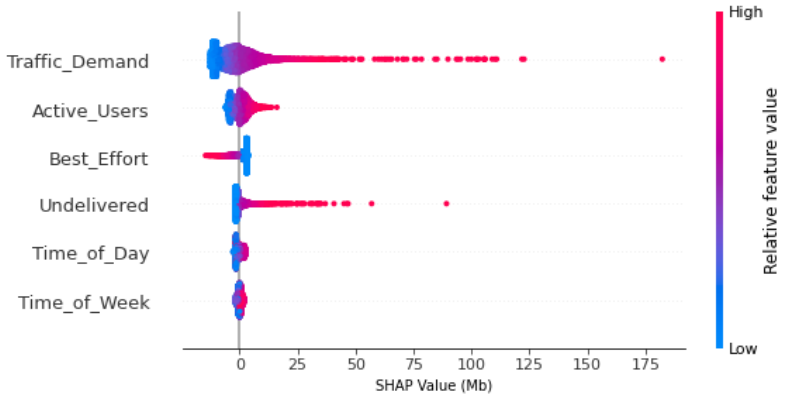
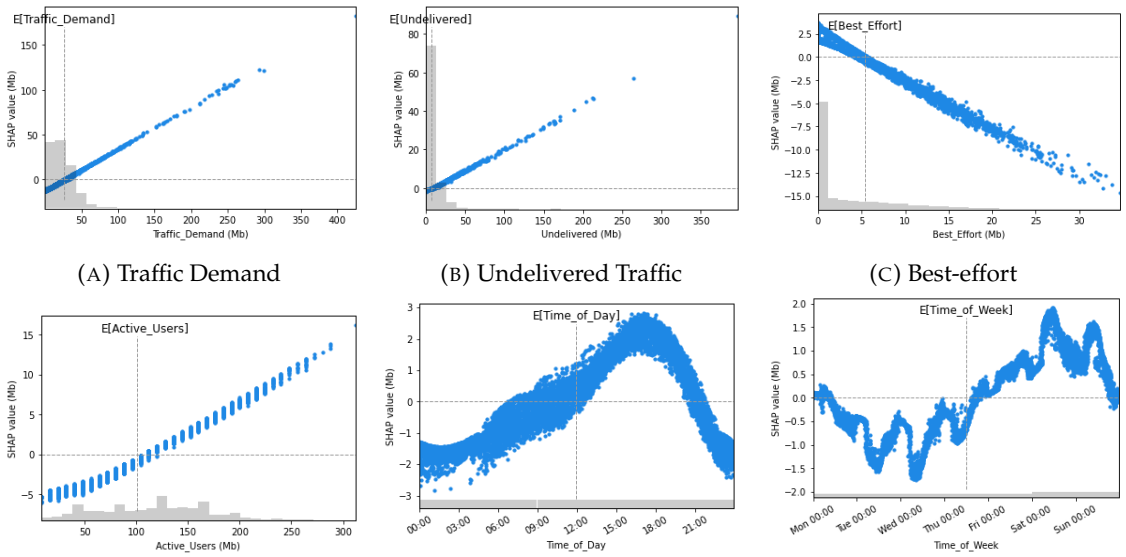


FIGURE 3.5: SHAP summary plot of our STRR model.

Importantly, SHAP summary plots can reveal a number of important trends about the model's general behaviour. For example, in Fig. 3.5, the summary plot for the past traffic demand reveals that low traffic demands generally result in lower reservations (i.e., blue dots are shown to be associated with negative SHAP values), while high demands tend to result in higher reservations (i.e., red dots are associated with positive SHAP values), with an inflection point occurring around the expected traffic demand (i.e., the purple region) etc. In general, trends can be identified for each of the features of the model and subsequently cross-referenced against the intuition of a domain expert to ensure only meaningful behaviours have been learnt by the model that will generalise to unseen environments. In the case of detecting unintuitive behaviours, additional considerations can be made to distinguish harmful behaviours from the possibility of new knowledge discovered by the model.

In Fig. 3.6, we show the SHAP dependence plots of each feature used by our STRR model when explaining samples from the test set. From these plots, a number of important insights can be gained about the model's general behaviour. Specifically, we see in Fig. 3.6 that the STRR model tends to increase its reservations monotonically as the feature values for the past-traffic demand, undelivered traffic and number of end users also increase, while we can also see that the model monotonically decreases its reservations as the best-effort amount increases. Moreover, in the case of the time of day feature, a strong cyclic relationship can be observed whereby the model tends to reduce its reservation amounts during the early/late hours of the day and increases its reservations during peak times in the afternoon/evening. Interestingly, a similar resemblance of this daily effect can also be seen in the time of week feature in addition to the model increasing its



(A) Traffic Demand (B) Undelivered Traffic (C) Best-effort

(D) Active Users (E) Time of Day (F) Time of Week

FIGURE 3.6: SHAP dependence plots of the STRR model. For a given subplot, each point corresponds to a single SHAP value computed for that feature based on one of the samples explained from the test set. For reference, we also show the general distribution of each feature (grey region) and its expected value (vertical grey line).

average reservation amounts on days, such as Friday, Saturday and Sunday (i.e., the end of the typical working week).

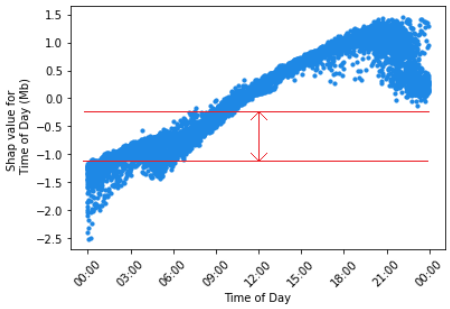


FIGURE 3.7: SHAP dependence plot revealing a concerning disparity associated with the start and end of each day for the time of day feature.

In the general context of resource management, global explanations can aid the HITL in identifying and debugging potential flaws in the model before its final deployment phase. As an example of how global insights were used for model debugging during the development of our own STRR model, we consider the SHAP dependence plot shown in Fig. 3.7, which corresponds to the time of day feature computed during the early design stages of our model. In contrast to the behaviour that this feature exhibits in our final model, i.e., as depicted in Fig. 3.6e, Fig. 3.7 reveals a significant disparity between the

start and end of each day, which we consider as concerning given that these two time stamps occur consecutively after one another.[7] In our final model, this problem was overcome by applying the 'trigonometric' transformation method described in the work of [158] to both the time of day and time of week features, as given by the expressions (3.8) and (3.9), respectively. Essentially, this transformation operates by splitting each feature into two new sub-features related to its sine and cosine components. Hence, the total number of features used by our final STRR model was eight features to represent the six main sources of information available to the slice tenant. We also note that the final SHAP values for the time of day and time of week features have been calculated by additively summing the individual SHAP values of their respective sub-features in our presented results. This additive approach was chosen based on the fact that our proposed framework necessarily considers AFA-based explanations in its design.

$$X_{Day} \longrightarrow sin\left(\frac{2\pi X_{Day}}{1440}\right), cos\left(\frac{2\pi X_{Day}}{1440}\right), \tag{3.8}$$

$$X_{Week} \longrightarrow sin\left(\frac{2\pi X_{Week}}{10080}\right), cos\left(\frac{2\pi X_{Week}}{10080}\right). \tag{3.9}$$

3.5.4 XAI Evaluation

As part of the final stage of our methodology, we propose the use of XAI metrics to evaluate the faithfulness and accuracy of the explanations produced by our framework. Here, we consider the following XAI metrics from [81] to serve as a starting point for achieving this task:

(a) *Local Accuracy:* We recall from Section 2.1.5, that this metric can be used to measure the ability of an AFA-based explanation method to produce importance scores ϕ_i that sum up to the original output of the model $f(\mathbf{x})$ for the sample being explained. Similar to [81], we evaluate the local accuracy across 100 random samples from the test set using equations (**??**) through (2.23) and compute the average score across all explanations.

(b) *Mask-Based Metrics:* From Section 2.1.5, we recall that the mask-based metrics refer to a suite of six different metrics that can be used collectively to assess the ability of

[7] We note that similar behaviour was also observed in the time of week feature.

an explanation method to identify features that increase or decrease the output of the model the most and those that have the largest impact on the model's accuracy. Moreover, each of these metrics results in a curve that can be used to cross-validate the relative quality of explanations produced under different XAI methodologies. However, beyond a relative comparison, these metrics lack an absolute scale or ground truth, which becomes essential for determining the true faithfulness of the explanations. In our work, we propose to augment the mask-based metrics with an absolute scale as follows:

(1) For a given explanation, we first compute each of the mask-based metrics as originally presented in [81]. We view the resulting curve across each metric as having a relative but unknown absolute quality, which numerically relates to the area that each curve occupies above the $y = 0$ line for each metric.

(2) For each metric calculated in step (1), the ground truth or absolute scale corresponds to the curve which occupies the largest or smallest area above the $y = 0$ line, depending on the metric being considered. Here, a crucial insight that we propose to leverage is that the resulting curve produced by each metric in step (1) depends solely on the relative rankings of the feature importance scores (for some metrics, the sign of the importance scores are also considered) but crucially never the raw values of the importance scores themselves. Thus, the optimal curve can be determined by exhaustively permuting across all possible combinations of feature importance rankings (and sign, where relevant) and selecting the permutation which yields either the highest or lowest Area Under the Curve (AUC) for the metric being considered.

(3) In order to arrive at a standardised score representing the faithfulness of the explanation in step (1), we compare the AUC of the explanation from step (1) with the AUC of its corresponding optimal score from step (2), where we define the AUC in this case to be the area above the $y = 0$ line *after* both curves have been *shifted down* by the minimum point occurring jointly along the y-axis of both curves. We note that this definition for AUC represents a generalisation of that previously proposed in [78, equation (12)], whereby an evaluation metric similar to the 'Remove Absolute' metric was presented.

Similar to [81], we compute each of the mask-based metrics across 100 random samples from the test set and compare the results against their ground truth in each case. In addition, we also calculate the mask-based metrics across a 'random-guessing' XAI approach, which essentially assigns zero-mean random importance scores to each of the features of a particular sample. We further take the average AUC across 100 randomised trials of each sample being assessed.

We summarise the final results of each of these metrics applied to our proposed methodology in Table 3.4. We also illustrate in Fig. 3.8, the resulting curves for the mask-based metrics computed under each of the three cases discussed above (e.g., STRR model, optimal, and random).

We recall that an optimal value for the 'Local Accuracy' metric occurs at a score of 1.0 while a score at or close to 0 signifies a highly inaccurate explanation method. Hence, we see from the first row in Table 3.4, that our methodology has achieved an optimal score of 1.0. While this initial result provides a strong early indication of the suitability of our methodology for use in resource management applications, such as STRR, we next consider the performance of our methodology across the mask-based metrics. In particular, we recall in the case of the 'Remove Negative Mask', 'Remove Absolute Mask' and 'Keep Positive Mask' metrics that a higher AUC implies greater faithfulness while a smaller AUC on the 'Remove Positive Mask', 'Keep Negative Mask' and 'Keep Absolute Mask' metrics imply greater faithfulness. Surprisingly, when compared against the AUCs obtained by the optimal and random cases, as shown by the last two columns of Table 3.4, we find that our proposed methodology achieves near-optimal performance across almost all the mask-based metrics, with the worst-achieving case ('Keep Absolute' metric) still yielding significant improvement over the random case. While these results provide a strong overall validation for our proposed XAI methodology under the STRR use case, we emphasise that, in a general real-world scenario for resource management, the ability to evaluate the accuracy and faithfulness of the underlying explanations serves as a vital component in allowing the HITL to both trust and act upon potentially crucial decisions made by the underlying AI/ML model.

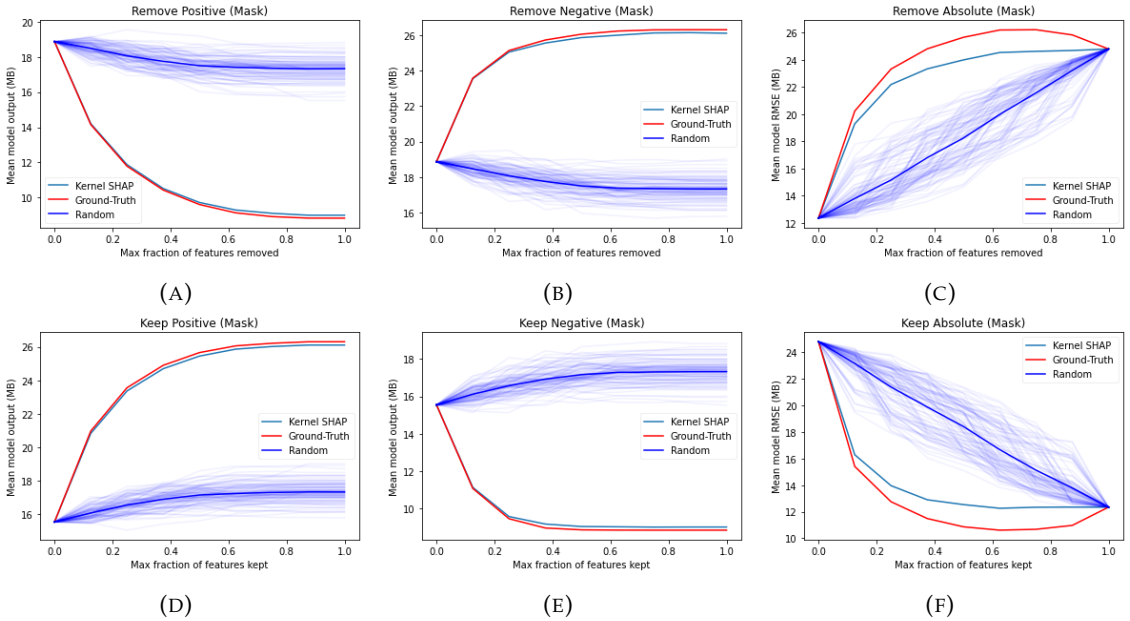


FIGURE 3.8: Performance curves of our proposed methodology across multiple XAI metrics. For each metric, we also show the optimal-achieving curve and curve of a random explanation method (averaged across 100 randomised trials).

TABLE 3.4: Summary of Evaluation Performance

Evaluation Metric	Model	Absolute	Random
Local Accuracy (δ)	1.0	1.0	-
Keep Positive*	68.77	70.2	10.66
Remove Negative*	62.04	63.18	3.28
Remove Absolute*	82.51	92.12	48.57
Remove Positive*	16.98	16.05	71.26
Keep Negative*	7.49	6.34	64.42
Keep Absolute*	26.32	16.45	62.14

* Measured in units of area.

3.6 XAI Evaluation: A General Complexity Analysis

In this subsection, we provide a brief overview of the computational complexity associated with the ground truth scores of the mask-based metrics previously presented in Section 3.5.4 as part of the evaluation of our proposed XAI methodology for resource management. Specifically, we begin our analysis by considering the 'Keep Positive' metric from [81]; following the outcome of our analysis, we discuss how the presented findings can be generalised among the remaining mask-based metrics.

3.6.1 Keep Positive Metric

In a general setting, the 'Keep Positive' metric can be summarised by the following sequence of steps:

1. Given an N-dimensional input sample, $\mathbf{x} \in \mathbb{R}^N$, and its associated local explanation, $\boldsymbol{\phi} \in \mathbb{R}^N$.

2. Create a new sample, $\mathbf{x}'$, where each feature in $\mathbf{x}'$ is "removed" from the model, i.e., by assigning each feature in $\mathbf{x}'$ with its average value across the train set (mean-masking).

3. Observe the output of the model, $y_i = f(\mathbf{x}')$, with $i = 0$.

4. Using the values stored in $\boldsymbol{\phi}$, identify the feature in $\mathbf{x}$ that has the most positive importance score (if any exists). Replace the value of this feature in $\mathbf{x}'$ with its corresponding value in $\mathbf{x}$ (i.e., unmask). The feature with the second most positive importance score in $\boldsymbol{\phi}$ becomes the most positive at the next repeat of this step.

5. Observe the output of the model, y_i, with $i \rightarrow i + 1$.

6. Repeat steps 4 and 5 until all possible features in $\boldsymbol{\phi}$ have been unmasked, i.e., until $i = N - 1$.

7. Plot y_i against the fraction of unmasked features, j, where $j \coloneqq i/(N-1)$ for all $i \in \{0, ..., N-1\}$.

As an important insight, we note that the above procedure only involves a comparison of the relative rankings and sign of each importance score while there is no further significance placed on the values of each importance score in $\boldsymbol{\phi}$ once they have been ranked according to their absolute magnitude and sign. Thus, in our analysis, we essentially aim to derive the complexity required to exhaustively examine all possible combinations of feature rankings and sign in order to determine the combination that yields the highest possible score on the 'Keep Positive' metric.

To aid our analysis, we consider a simplified scenario where $\mathbf{x}$ comprises only 3 features, i.e., $\mathbf{x} \in \{A, B, C\}$. In this case, there are exactly 3! possible ways in which all three features can be re-arranged and ranked according to their absolute magnitude, as

illustrated by expression (3.10). In general, there are N! possible orderings for an input sample with N positive features, i.e., $\mathbf{x} \in \mathbb{R}+^{N}$.

$$3! \left\{ \begin{array}{c} A, B, C \\ A, C, B \\ B, A, C \\ B, C, A \\ C, A, B \\ C, B, A \end{array} \right\} \tag{3.10}$$

In the case where the features in $\mathbf{x}$ are also ranked according to their sign, the total number of possible orderings increases to $3!\,2^3$. This can be explained by the fact that each initial ordering in expression (3.10) now effectively expands into 2^3 additional possible combinations, as illustrated by expression (3.11), where features with bars have a negative sign while features with no bars have a positive sign. In general, there are $N! \times 2^N$ possible orderings that can obtained for an input sample $\mathbf{x} \in \mathbb{R}^N$ when ranked by absolute magnitude and sign.

$$3! \left\{ \begin{array}{l} A, B, C \rightarrow \left\{ \begin{array}{ccc} \overline{A}, & \overline{B}, & \overline{C} \\ & \vdots & \\ A, & B, & C \end{array} \right\} 2^3 \\ \quad \vdots \\ C, B, A \rightarrow \left\{ \begin{array}{ccc} \overline{C}, & \overline{B}, & \overline{A} \\ & \vdots & \\ C, & B, & A \end{array} \right\} 2^3 \end{array} \right. \tag{3.11}$$

Total complexity = $3! \times 2^3$

Importantly, as the 'Keep Positive' metric only considers the rankings of features that have a positively associated importance score, while effectively ignoring the rankings of negatively associated features (i.e., step 4 above), the resulting complexity can, in fact, be reduced to the order of $3! \times (3+1)$, as illustrated by expression (3.12). In this expression, neighbouring features that share a negative sign are essentially treated as a single entity, thus reducing the total number of combinations that need to be considered.

$$3!\left\{\begin{array}{l} A,B,C \rightarrow \{\overbrace{\;A,B,C \quad \overline{A},B,C \quad \overline{AB},C \quad \overline{ABC}\;}^{3+1} \\ \quad\vdots \\ C,B,A \rightarrow \{\underbrace{\;C,B,A \quad \overline{C},B,A \quad \overline{CB},A \quad \overline{CBA}\;}_{} \end{array}\right. \tag{3.12}$$

$$\text{Total complexity} = 3! \times (3+1)$$

In the general sense, the complexity indicated by expression (3.12) can be summarised as comprising $N! \times (N+1)$ permutations. However, this complexity figure can also be further reduced by carefully identifying additional redundancies in expression (3.12). To elaborate on this, expression (3.13) shows an exhaustive layout of expression (3.12). From this, it can be seen that some additional redundancies occur whenever the same set of negative features repeats itself along a certain column. At the bottom of expression (3.13), we show the total number of redundant elements along each column as well as the general trend that captures how this amount varies along each column (in the case of the number of non-redundancies).

$$\left\{\begin{array}{cccc} A,B,C & \overline{A},B,C & \overline{AB},C & \overline{ABC} \\ A,C,B & \overline{A},C,B & \overline{AC},B & \overline{ACB} \\ B,A,C & \overline{B},A,C & \overline{BA},C & \overline{BAC} \\ B,C,A & \overline{B},C,A & \overline{BC},A & \overline{BCA} \\ C,A,B & \overline{C},A,B & \overline{CA},B & \overline{CAB} \\ C,B,A & \overline{C},B,A & \overline{CB},A & \overline{CBA} \end{array}\right\} \tag{3.13}$$

Number of redundancies:	**0**	**0**	**3**	**5**
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
Number of non-redundancies:	${}^3P_3$	${}^3P_2$	${}^3P_1$	${}^3P_0$

For the 'Keep Positive' metric, the exact number of permutations required, in the general setting, can be captured by the expression,

$$\sum_{i=0}^{N} {}^N P_i \tag{3.14}$$

where N is the number of features in the explanation and ${}^{\{\cdot\}}P_{\{\cdot\}}$ refers to the permutation operator. Finally, since the big-O complexity of the summation operator in expression (3.14) can be given by $\mathcal{O}(N)$, while the complexity of the permutation operator can be given by $\mathcal{O}(N!)$ [159], the overall complexity of the 'Keep Positive' Metric can be summarised as $\mathcal{O}(NN!)$.

3.6.2 General Complexity of Mask-based Metrics

The general complexity required for computing the ground truths of the remaining mask-based metrics can be determined by carefully considering the similarities and differences shared between the remaining metrics and the 'Keep Positive' metric. In particular, we note that all of the 'Remove Positive', 'Remove Negative' and 'Keep Negative' metrics share the same complexity as with the 'Keep Positive' metric, i.e., $\mathcal{O}(NN!)$, since all four of these metrics operate on a similar basis whereby the absolute magnitude and signs of the features importance scores are jointly considered during their calculation. On the other hand, the 'Keep Absolute' and 'Remove Absolute' metrics only consider the absolute magnitudes of the features in their ranking process, and therefore, can be computed at a reduced (but still large) complexity given by that of expression (3.10), i.e., $\mathcal{O}(N!)$. In summary, computing the ground truths of each of these metrics requires considerable computational resources that grow at an exponential rate as the number of features in the explanation increases. In a practical setting, this may potentially limit the total number of features that can considered by the associated AI/ML solution to below tens of features. In Chapter 6, we later discuss briefly some potential solutions that may help overcome this challenging drawback.

3.7 Chapter Conclusion

In the introduction of this chapter, we introduced the important problem of resource reservation within a sliced network environment and subsequently highlighted the findings following an initial study in which two deep-learning solutions comprising both a vanilla DNN and LSTM architecture were proposed to resolve the reservation task from a tenant's perspective. Importantly, the results of this initial study demonstrated the motivation and feasibility in adopting deep-learning solutions for resource reservation within

sliced networks; in this case, it was shown that both deep-learning solutions were able to significantly outperform an ARIMA baseline at multi-step-ahead reservations, achieving significantly lower MSE than the baseline, particularly as the prediction horizon grew larger.

Following from this, the remainder of this chapter introduced and explored the important concept of explainability of complex AI/ML models used in resource management problems, such as for STRR. Importantly, our work in this space has been motivated by the fact that many AI/ML models, such as DNNs, are commonly perceived as "black-boxes" due to their inherent lack of transparency in their decision-making processes and behaviour [160]. In the context of resource management, and in particular, the problem of resource reservation, this lack of transparency translates to severe potential risks to both the reliable operation of the network, as well as revenue loss for operators should their reservation models, which have been entrusted to make autonomous real-time leasing decisions, fail and behave in an unexpected manner during their operation.

To help alleviate these risks and overcome the black-box limitation of AI/ML solutions, we have proposed a methodology based on state-of-the-art XAI techniques to explain and enhance the transparency of black-box models employed in resource management tasks for sliced networks, such as STRR. In particular, our proposed methodology allows operators to understand and analyse the behaviour of their models from both a local and global perspective. Specifically, we have demonstrated, using our own STRR model as an example use case, how the insights from our methodology can be used to monitor the real-time decisions of the model, to reveal global trends about the model, as well as aid in the diagnosis of potential flaws in the model's decision-making process. In addition, and as part of our proposed methodology, we have also presented a procedure that can be used to quantitatively validate the faithfulness of the explanations based on an extensive range of XAI metrics. Our presented evaluation procedure includes the calculation of ground truths for each metric for which we have further analysed the complexity of this procedure.

Overall, the findings of this chapter provide an initial and crucial insight into our first research question, RQ1, which aims to understand how XAI should be best integrated into high-stakes areas of the network domain. In the first instance, our proposed methodology can allow operators and other stakeholders of the network to improve the overall

performance and reliability of their models, before they are deployed into a real-world environment, as well as to continue monitoring their decision-making actions once they have been deployed. Additionally, to the best of our knowledge, our work also presents the first attempt to employ the use of quantitative XAI metrics with a proposed ground truth that allows the faithfulness of the explanations to be assessed within the context of network slicing, and possibly even the general field of telecommunications. By enabling operators and stakeholders to assess and validate the faithfulness of the explanations that they are presented with, this final step ensures that our methodology can be trusted and remains actionable in a real-world context. This concludes the results and findings towards our first contribution, C1.

In the next chapter, we turn to our second use case explored within this thesis, namely, the problem of network intrusion detection, and present our second contribution, C2, which aims to provide a direct answer to our third research question, RQ3, and partially contribute towards our overarching research question, RQ1.

Chapter 4

Robust Network Intrusion Detection through XAI

4.1 Introduction

In this chapter, we present our second contribution, C2, which partially addresses RQ1 and directly addresses RQ3. Specifically, we recall from Section 1.3, that RQ1 relates to the integration of XAI into high-stakes areas of the networking domain to enhance the overall reliability and service of the network, while RQ3 relates to the idea of reducing the effort required by the HITL to manually analyse the information contained within the explanations presented to them, while also enhancing the overall performance of the system. In this chapter, we aim to address RQ1 and RQ3 by considering our second use case scenario; the problem of network intrusion detection. As previously discussed in Section 1.3, this particular area of research presents a number of natural motivations for its applicability towards RQ1 and RQ3. Namely, in a typical intrusion setting, complex AI/ML models are expected to monitor vast quantities of network traffic data and to make timely responses to potential intrusions [161]. Moreover, as each alert from the intrusion system potentially requires further inspection from cybersecurity personnel, who must carefully examine the decisions of the model as well as their corresponding explanations to determine the necessary actions to take, so does the problem of missed attacks due to alert fatigue or analyst burnout become crucially relevant [162], [163]. In many cases, the potential impact caused by missed attacks can have catastrophic implications toward increased network downtime as well as severe financial and reputational consequences for the organisations and their personnel who fall victim to the attack, further

calling for these systems to have high accuracy, and in particular, robustness to zero-day attacks.

In this chapter, we aim to combat these challenges by proposing the idea of passing explanations of an intrusion detection model as input into additional AI/ML models for further processing. More specifically, we propose to combine the benefits of XAI alongside the strengths of both supervised and unsupervised NIDSs, and propose a five-stage pipeline for robust network intrusion detection. During the first two stages of our proposed pipeline, we train a supervised intrusion detection model to act as a "front-end" classifier for detecting network attacks. During the third stage of our pipeline, we leverage the prominent SHAP framework to devise explanations of our model. In the fourth stage, we then pass these explanations as input into a deep autoencoder model, whose primary purpose is to learn a latent representation of the initial model's 'typical' behaviour seen during its training phase. Based on the hypothesis that our model will only deviate from this typical baseline when trying to classify unseen or zero-day attacks, we then perform anomaly detection based on the reconstruction error of the autoencoder to determine whether a particular traffic flow *potentially* belongs to a new class of attacks during its testing phase.[1] Finally, we combine the outputs from the second (i.e., front-end model), third (i.e., SHAP explanation) and fourth (anomaly detection) stages to arrive at an enhanced prediction for detecting attacks on the network and for explaining the reasons behind the attack. Additionally, we note that while the first three stages of our pipeline bears similarity to existing works for explainable intrusion detection, the use of explanations for anomaly detection in the later stages of our pipeline appears to be almost completely unexplored in current research (we refer the reader to Section 2.2.2 for more details).

To the best of our knowledge, this work is the first to expand beyond the mere use of XAI as an E2E tool to aid the HITL in understanding why decisions are made by the NIDS, but also to explore the feasibility of employing additional ML mechanisms on these explanations to further enhance the performance of the NIDS. Moreover, the results following an extensive evaluation conducted on the NSL-KDD intrusion dataset

[1]It is important to note that traffic may still be deemed anomalous even if it does not stem from a new class of attacks. Such cases, which we refer to as new 'normal', may occur if the training data is incomplete or unable to account for all conceivable input-output scenarios.

[164] show that our solution is able to outperform various state-of-the-art works in terms of its overall accuracy, recall and precision.

The remainder of this chapter is organised as follows: in Section 4.2, we outline the details of our proposed methodology for enhancing intrusion detection performance through the use of XAI and additional AI/ML. In Section 4.3, we present the implementation details of our proposed methodology, while in Section 4.4, we discuss our evaluation strategy and subsequently present our results. In Section 4.5, we offer our conclusions for this Chapter.

4.2 Proposed Methodology

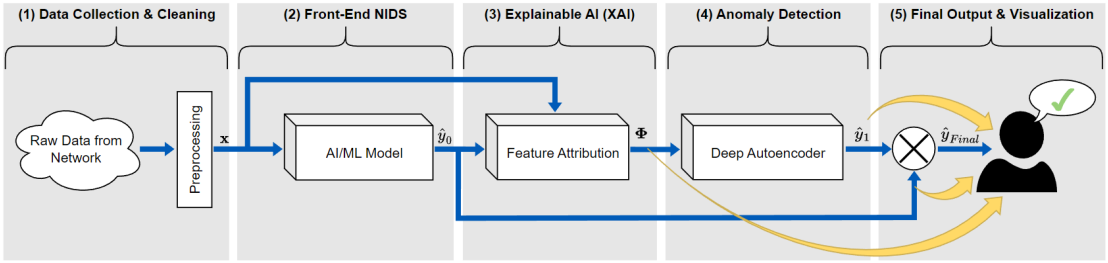


FIGURE 4.1: General overview of our proposed methodology.

As previously mentioned, our proposed methodology is motivated by the idea of leveraging XAI and additional ML to automatically improve the performance of a NIDS. In Fig. 4.1, we illustrate the general architecture of our proposed methodology, which comprises five main stages: i) *Data Collection & Cleaning*: collects raw information from the network and preprocesses it into suitable train/test samples to be classified by the front-end NIDS; ii) *Front-End NIDS*: uses AI/ML to perform preliminary intrusion detection of the incoming network traffic (binary classification); iii) *Explainable AI:* implements feature attribution-based XAI to explain the behaviour of the front-end NIDS; iv) *Anomaly Detection*: provides an additional layer of anomaly detection by leveraging a Deep Auto-encoder (DAE) (unsupervised ML) trained on explanations from the training data to distinguish between known behaviours of the front-end NIDS and unknown behaviours encountered during its testing/online deployment; v) *Output Update & Visualization:* combines the output of the front-end classifier and anomaly detection stages

together to obtain an enhanced prediction of the network state, while also supporting further analysis of the system's behaviour by a HITL. In what follows, we elaborate further on each of these main stages of our methodology.

4.2.1 Data Collection & Cleaning

During the initial stages of our methodology, raw data is first collected from the network through suitable network analysers/deep packet inspection tools and then passed through a data cleaning and preprocessing stage. Here, the primary purpose of the preprocessing stage is to perform any necessary steps required to transform and prepare the raw data into discrete samples, which can be easily and efficiently digested by the front-end NIDS located downstream in our methodology. For example, if we consider a typical scenario where the raw data might consist of various Packet Capture (PCAP) files that have been collected from the network, while a neural network or tree-based model is used as the basis of the front-end NIDS, then the preprocessing stage may involve the following sequence of sub-steps:

i) **Feature Extraction:** Before data-driven decisions can be made about the network, suitable features are first extracted from the raw PCAP files to form the basis of the training/testing samples needed by the front-end NIDS. For this task, network analysers, such as SiLK [165], Tranalyzer [166], CICFlowMeter [167], etc., may be employed to generate bidirectional flows of the given packet traces, and to extract various features from each flow, including, for example, the IP addresses of the source/destination pair, the ports used by the source/destination, the amount of time the flow has been active, the average packet payload across the flow, etc.

ii) **Dataset Cleaning:** Following the extraction of an initial set of training/test samples, the samples require a series of additional cleaning measures before a final dataset can be obtained to train/test the front-end NIDS efficiently. For example, typical cleaning measures that may be applied to the training data include removing duplicate samples from the initial training set, removing redundant or irrelevant features from the final feature set (i.e., feature selection), numerically encoding any categorical features (e.g., transport protocol used, traffic service type, connection state, etc.), and scaling/normalizing features to aid the training of the front-end

NIDS. During the testing and online deployment of the model, similar preprocessing steps are then re-applied to the incoming test samples, taking care to ensure any hyperparameters (i.e., scaling factors, encoding parameters, etc.) used during this stage are derived solely from the training data to prevent any potential data leakage from occurring [168].

4.2.2 Front-End NIDS

Samples emerging from the preprocessing stage are passed as input into the front-end intrusion detection model, as well as the XAI module. Here, the purpose of the front-end NIDS is to provide a preliminary (binary) prediction of whether or not an attack is occurring within the network, while the purpose of the XAI stage is to produce a corresponding explanation outlining the factors that have led to the front-end NIDS's decision. Formally, let $f : \mathbb{R}^N \rightarrow \mathbb{R}^1$ denote the mapping function implemented by the front-end NIDS. If we consider a single sample of data passed as input to the front-end NIDS, i.e., denoted by the vector, $\mathbf{x} = [x_1, ..., x_i, ..., x_N]$ with x_i representing the i^{th} feature in $\mathbf{x}$, then the corresponding output, $\hat{y}_0 \in \mathbb{R}^{[0,1]}$, is assumed to be a real-valued scalar corresponding to the estimated probability of the flow being malicious, with $\hat{y}_0 \geq 0.5$ implying malicious traffic and $\hat{y}_0 < 0.5$ implying benign traffic.

In practice, a wide variety of ML/AI algorithms can be used to implement the function of the front-end NIDS in our methodology. In this work, we utilise an XGBoost model to serve as the front-end detector. This type of model has become widely popular within the cybersecurity space in recent years due to its ability to accurately detect known attack types and also shares the property of having a complex internal structure which further warrants the need for additional explainability measures alongside its application.

4.2.3 Feature Attribution (XAI)

Following the prediction of the front-end classifier, the third stage of our methodology involves the use of XAI methodologies to explain the behaviour of the front-end NIDS. Specifically, in our current work, we focus primarily on explanation techniques that adhere to the general class of AFA, since these types of explanations have become widely

adopted throughout the cybersecurity space and allow for intuitive explanations that can be easily scrutinised by human analysts (for in-depth examples, we refer the reader to the discussion in Section 2.2.2).

Formally, when presented with a particular input, $\mathbf{x}$, and corresponding output prediction from the front-end model, $\hat{y}_0$, the feature attribution stage of our methodology produces a corresponding explanation in the form of a vector of "feature importance scores", given by $\boldsymbol{\phi} = [\phi_1, ..., \phi_i, ..., \phi_N]$, in which each element, ϕ_i, corresponds to the impact that feature x_i has on the observed output, $\hat{y}_0$. In general, any feature attribution method capable of mapping each ϕ_i to either a positive or negative value, i.e., thereby allowing the *magnitude* as well as the *direction* in which each feature affects the decision of the model to be conveyed in the explanation, may be implemented as part of the third stage of our methodology (e.g., SHAP, LIME, etc.).

In this work, we specifically explore the performance our methodology when the third stage is implemented using the 'TreeExplainer' method from [81], which has recently been proposed as a state-of-the-art approach for obtaining accurate SHAP explanations of tree-based models within polynomial time complexity. In addition to allowing both the magnitude and direction of each feature's impact on the model to be shown, we recall from Section 2.1.4 that the SHAP values take their basis from the Shapley value from game theory, and therefore, exhibit a number of desirable properties in contrast to most other XAI methods in the literature [81]. For example, one key property of this approach, known as the "local accuracy" property, ensures that the sum of all SHAP values for a specific instance adds up to the *difference* between the model's output for that instance and a constant baseline, ϕ_0 :

$$\sum_{i=1}^{N} \phi_i = \hat{y}_0 - \phi_0. \tag{4.1}$$

In 'TreeExplainer', the baseline value, ϕ_0, corresponds to the average or expected output of the front-end model seen during training, and can be interpreted as the initial output of the model before the impact of any features are taken into account. Moreover, we see from the right-hand side of equation (4.1), that the explanation essentially *decomposes* the output of the model (ignoring the constant baseline effect for emphasis)

amongst each of the input features. In the context of our NIDS, this means that cybersecurity experts can easily understand the relative importance of each feature as it increases or decreases the probability of a flow being classified as malicious. This, in turn, can shed much-needed light on the nature of the attack, as well as how best to counteract it.

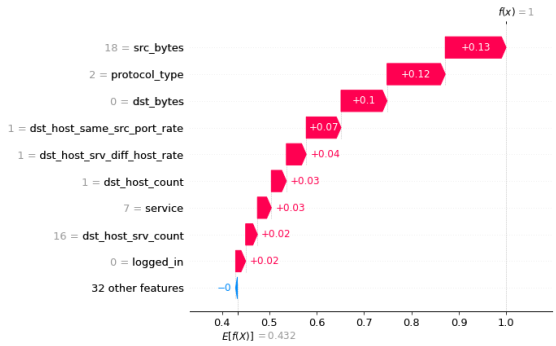


FIGURE 4.2: Example SHAP explanation from the NSL-KDD training set.

As an example, Fig. 4.2 illustrates how the SHAP technique [54] can be used to explain a single training sample from the NSL-KDD dataset used in our work. In this particular example, the occurrence of a probe attack in the network has led our front-end NIDS to correctly classify the sample as being malicious with a probability of 1.0. In the corresponding explanation, features that have a positive impact on the model's decision are shown next to red bars, while features that reduce the output of the model are shown next to blue bars. In this example, the baseline value, ϕ_0, corresponds to approximately 0.432, while the features corresponding to the number of bytes sent from the source to the destination ('src_bytes') and the connection protocol type ('protocol_type') are identified as the main features that have led the model to increase its predicted probability for an attack. Moreover, other features, such as the number of bytes sent from the destination to the source ('dst_bytes'), the percentage of connections sent to the host using the same source port ('dst_host_same_src_port_rate'), etc., are also indicated as having minor positive contributions towards the model's decision, while the remaining 32 features used by the model (aggregated together in the shown explanation) have a combined negative but almost negligible impact on the model's decision.

4.2.4 Anomaly Detection

During the first stages of our pipeline, the use of a supervised ML solution as part of the front-end NIDS implies that we can generally expect our system to achieve good performance on previously seen attacks, but still face the likely possibility of performing poorly against unseen attacks. In the fourth stage of our methodology, we attempt to address this shortcoming by exploring the hypothesis that our model will behave differently when attempting to classify traffic flows emanating from zero-day attacks, compared to how it behaves when classifying traffic similar to that seen during training. To test this hypothesis, we turn to the discriminatory power provided by unsupervised learning techniques, such the DAE. Specifically, we design an anomaly-based NIDS, where the aim in this case is to accurately differentiate between previously seen behaviours (ie., explanations based on the training data), and new behaviours (i.e., arising from zero-day attacks, or possibly, from new 'normal' traffic). Below, we discuss in more detail the general operation of this stage.

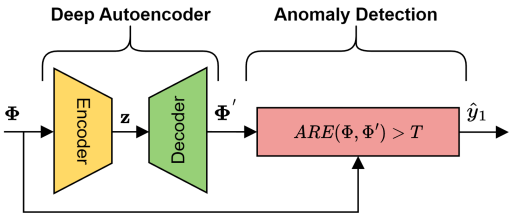


FIGURE 4.3: Detailed outline of Anomaly Detection stage.

In Fig. 4.3, we provide an expanded overview of our fourth stage, which ultimately performs the role of unsupervised anomaly detection. Here, explanations arriving from the previous stage are first passed through a DAE, which essentially comprises two back-to-back DNNs, where the task of the first DNN (i.e., the "encoder") is to map the input explanation into a latent representation, while the task of the second DNN (i.e., the "decoder") is to decode the latent representation back into an accurate reconstruction of the original explanation without requiring any additional label information (i.e., unsupervised learning).

More formally, let $\mathbf{z} \in \mathbb{R}^M$ be a vector denoting the latent output of the encoder stage, which consists of M real-valued elements. Here, we are particularly interested in the *undercomplete* case, where the output dimension of the encoder is constrained to be much smaller than its input dimension, i.e., with $M \ll N$, which ensures the encoder

learns a compressed latent representation of the original input using only the most salient aspects of the explanations encountered during its training [37]. On the other hand, let $\boldsymbol{\phi}'$ denote the reconstructed explanation at the output of the decoder, which we optimize jointly alongside the encoder to accurately reconstruct explanations across the training distribution, according to the objective function:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{\phi}) \sim \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\phi}'), \tag{4.2}$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation function, $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}$ denotes the empirical distribution of explanations used to train the autoencoder, $\mathcal{L}$ denotes the loss function that is minimized during the training process and measures the dissimilarity between the original and reconstructed explanation (e.g., typically L2 or L1 loss), while $\boldsymbol{\theta}$ denotes the parameters of the autoencoder that are optimized (i.e., neural weights and biases of the encoder and decoder stages, etc.).

Importantly, we aim to capitalise on the autoencoder's natural ability to perform anomaly detection by identifying any instances where abnormal behaviour of the front-end NIDS occurs during its online operation. To accomplish this, we calculate the Absolute Reconstruction Error (ARE) between the original and reconstructed explanations during testing, where,

$$ARE(\boldsymbol{\phi}, \boldsymbol{\phi}') := \frac{\sum_{i=1}^{N} |\phi_i - \phi_i'|}{N}. \tag{4.3}$$

Since a well-trained autoencoder can be expected to accurately reconstruct input explanations that are similar to the training distribution, $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}$, but perform poorly when attempting to reconstruct inputs that are not, we assume any flows with an ARE greater than some threshold, T, as being anomalous. Moreover, to determine a suitable value for T, we first apply our methodology across the training distribution, $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}$, to obtain a corresponding ARE distribution, $\hat{\mathcal{E}}_{\boldsymbol{\phi}^{(train)}}$. A suitable value for T is then derived by selecting a value that offers a good balance between the expected false positive and false negative detection rates, such as the 95th percentile of $\hat{\mathcal{E}}_{\boldsymbol{\phi}^{(train)}}$. The final output from the fourth stage, $\hat{y}_1$, can then be summarised according to:

$$\hat{y}_1 = \begin{cases} 1 \rightarrow \text{Anomaly}, & ARE(\boldsymbol{\phi}, \boldsymbol{\phi}') > T \\ 0 \rightarrow \text{Normal}, & \text{else} \end{cases} \tag{4.4}$$

4.2.5 Final Output & Visualization

In the final steps of our methodology, the predictions from both the front-end detector and anomaly detection stage are combined into a single scalar, $\hat{y}_{Final}$, representing the overall probability of an attack in the network, where

$$\hat{y}_{Final} = max(\hat{y}_0, \hat{y}_1). \tag{4.5}$$

The final output, $\hat{y}_{Final}$, is then presented along with the explanation, $\boldsymbol{\phi}$, and intermediate outputs, $\hat{y}_0$ and $\hat{y}_1$, to cybersecurity personnel for further analysis. Although specific details around such analysis fall outside the scope of this article, we briefly highlight two edge cases that may occur in a practical scenario. For example, in the case of a flow that has been deemed anomalous by the fourth stage but not the front-end NIDS, an analyst may have to manually investigate the flow and its explanation to determine whether it corresponds to a new attack or a sample that might be under-represented in the original training set, and therefore might require additional targeted re-training of the front-end NIDS. In cases where a flow is found to be malicious by the front-end NIDS but not by the fourth stage, it is likely that this type of flow stems from a previously seen or known attack, implying that an automated response might already exist and that further analysis from an analyst may not be required. Moreover, we stipulate that the former case may yield further benefits to an online-learning scenario, where the NIDS is continuously adapted over time to maintain its accuracy, while the combination of both these cases may have overall consequences related to the 'zero-touch' paradigm envisioned for 6G and next-generation networks [169], which we hope to explore further in future work.

4.3 Implementation Details

This section details the implementation aspects of our proposed methodology for robust intrusion detection, which was subsequently used to generate the results presented

in Section 4.4. Specifically, subsection 4.3.1 provides a brief overview of the NSL-KDD dataset, while subsection 4.3.2 discusses the data preprocessing steps that were applied to the NSL-KDD dataset used by the front-end NIDS and input explanations used by the DAE as part of the fourth stage of our methodology. In subsections 4.3.3 and 4.3.4, we discuss the implementation aspects of the front-end NIDS and XAI stages of our methodology, respectively. Finally, subsection 4.3.5 provides details on the DAE solution that was used to perform anomaly detection as part of the fourth stage of our proposed methodology.

4.3.1 NSL-KDD Dataset

Proposed by [164], the NSL-KDD dataset presents an improved version of the KDD'99 dataset, originally released by DARPA as one of the first publicly-available intrusion datasets to contain a wide host of realistic attacks to a military network. Similar to the KDD'99 dataset, the NSL-KDD dataset contains 41 network-related features derived from TCP/IP dumps, as well as offering examples of 23 different attacks in its training set, with an additional 17 new attacks across its test set. In comparison to KDD'99, the NSL-KDD dataset offers a number of important improvements aimed at promoting greater consistency and fairness when comparing across different NIDSs, including the removal of duplicate flows and the use of a proportional inclusion policy to minimise class-imbalance issues associated with rare attack types. Although the NSL-KDD dataset is limited in capturing examples of more modern attack types, we consider its use relevant to our work as it remains one of the few publicly available intrusion datasets suitable for evaluating the performance of a NIDS under a shift in training and testing distributions; in this work, we use the 'KDDTrain+' and 'KDDTest+' datasets for training and evaluating our solution, respectively.

4.3.2 Data Preprocessing

As previously outlined, our XGBoost model for supervised intrusion detection requires all inputs to the model to be of numerical modality. To ensure this requirement was met during the training and testing of our solution, we applied label encoding to all non-numerical features passed to our model using the 'LabelEncoder' method from the

Scikit-Learn [170] library for Python. For the NSL-KDD dataset, this includes three features which are initially categorical in format: 'protocol_type', 'service', and 'flag'. In the case of our DAE solution, all input features were converted into the range (-1, 1) using the 'MinMaxScaler' method from Scikit-Learn. Finally, as we only consider binary classification in our current work, we merge all attack labels in the NSL-KDD dataset into a single class of attacks.

4.3.3 XGBoost Front-End NIDS

We implement our XGBoost model using Scikit-Learn's ML API. To train our model, we use a binary logistic loss function and keep all remaining parameters of the model at their default values.

4.3.4 SHAP-based XAI

We use an open-source implementation of the 'TreeExplainer' method provided by the original authors in [81] to compute SHAP explanations of our front-end model. This method is characterised by a 'feature perturbation' hyperparamter, which specifies the type of perturbation process to use when masking features during the computation of their corresponding SHAP values. In this work, we set this hyperparamter to 'interventional' to ensure that the resulting explanations remain faithful to the model's true behaviour, as suggested by [80].

4.3.5 DAE-based Anomaly Detection

We implement both the encoder and decoder networks of our overall DAE model using the TensorFlow API [154]. Before training each network, we split the original training set into two subsets, and use roughly 80% of the original data as a new training set, and the remaining 20% as a validation set. We note that any samples used during the evaluation of our solution are kept separate from the training and validation sets used at this stage.

For our encoder network, we use 4 neural layers, including an input layer, two hidden layers, and an output layer. For each layer, we use a ReLU activation function with 41, 1456, 724, 14 neurons, respectively. In order to avoid over-fitting during training, we also apply dropout with strength 0.2 on the first hidden layer. Similarly, our decoder

consists of 4 layers with 14, 632, 1644, 41 neurons, respectively, as well as a ReLU activation function at each layer, except for the final output layer, which uses a tanh function instead to ensure all outputs fall within the range $(-1, 1)$. We train the encoder-decoder networks jointly for 1000 epochs using Tensorflow's default 'Adam' optimiser with a mean absolute error loss function, and an early-stopping criteria based on the validation loss during training. We also employ mini-batching of size 512 to aid the convergence speed of our autoencoder during training. In addition, we note that the final parameters of our encoder-decoder network have been chosen based on a mixture of empirical experiments, as well as a grid search strategy.

4.4 Evaluation

In this section, we provide an overview of our evaluation strategy (Section 4.4.1), followed by a summary and discussion of our results (Section 4.4.2).

4.4.1 Evaluation Strategy

For our evaluation, we consider separately the performance of: i) our supervised NIDS; ii) our anomaly-based NIDS; as well as iii) the combination of both NIDS in our overall pipeline. For cases (i) and (iii), we calculate performance metrics based on the general ability of our NIDSs to detect attacks, new or old, while for case (ii) we calculate metrics based solely on instances of new attacks. In our analysis, we consider common network intrusion metrics, such as accuracy, recall, and precision (for further details on each of these metrics, we refer the reader to Section 2.1.2). To benchmark our work, we also compare our results against those reported in various state-of-the-art works which also incorporate the NSL-KDD dataset in their evaluations [140] [171], as well as any other works which offer explainability in their solutions [144], [172]. We note that the results reported in [146] are not included in this comparison, as these only consider a portion of the NSL-KDD test set.

4.4.2 Results & Discussion

Table 4.1 summarises the performance of our supervised front-end NIDS, our anomaly-based NIDS, as well as our overall pipeline, where for simplicity, we assume any instance flagged as either malicious or anomalous to be an attack.[2] Here we see that, on its own, our supervised NIDS achieves a reasonable accuracy of 79% against general attacks, with an additional high precision of 96% but low recall of 65%. On the other hand, our anomaly-based NIDS achieves a relatively low precision of 43% on new attacks[3] but high recall of 89%, with an overall accuracy of 78%. Remarkably, when we combine both NIDSs together, our overall pipeline performs strongly across all three metrics, achieving an overall accuracy of 93%, precision of 91% and recall of 97%. This significant jump in performance supports our main hypothesis, that the explanations from our supervised NIDS can in fact be used alongside additional ML mechanisms to greatly enhance the performance of our intrusion system.

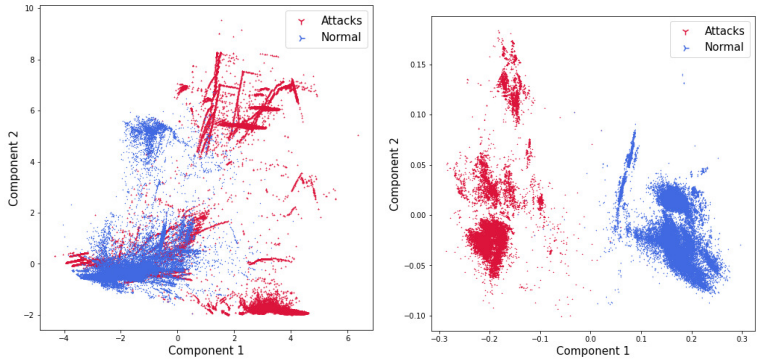
TABLE 4.1: Performance of our proposed NIDS pipeline.

Method	Accuracy (%)	Recall (%)	Precision (%)	Scope
Supervised NIDS	79.20	65.61	96.82	All
Anomaly-based NIDS	78.53	89.41	43.01	New
Overall Pipeline	93.28	97.81	91.05	All

To try and explain why this is the case, we apply Principle Component Analysis (PCA) to both the raw training data as well as its explanations. As seen in Fig. 4.4a, plotting the first 2 PCA components of the raw training data reveals a strong overlap between samples of normal and malicious traffic. On the other hand, a plot of the first 2 PCA components based on the training data's explanations shows that the 'explanation domain' is able to distinctively separate clusters of normal flows from malicious ones. We believe one possible reason is that the explanation model itself is formulated in terms of a simplified linear model, i.e., in generating the SHAP values, we inherently disentangle some of the non-linearities existing throughout the decision space. Additionally, when we examine the cumulative PoV explained by each PCA component, as shown in Fig. 4.5, we find that only 32% of the data variance is explained by the first 2 components of the raw training data, compared to 88% in the case of the explanations. As the PoV

[2]Since it is possible for an analyst to further validate each anomalous instance as being either an attack or new 'normal' instance, the results presented here constitute a lower bound on what may be achieved in practice.

[3]We believe this low precision can be largely attributed to the fact that all anomalies are assumed to be attacks, i.e., the precision is reduced due to the presence of new 'normal' instances being flagged as attacks.



(A) (B)

FIGURE 4.4: PCA applied to the training data (a), and the explanations (b).

can be regarded as a measure of how much of the original signal information is contained within each component, this suggests that the explanation domain is also capable of compressing greater amounts of important information (i.e., salient aspects from the decision boundary between normal and abnormal traffic) more efficiently than the raw training data.

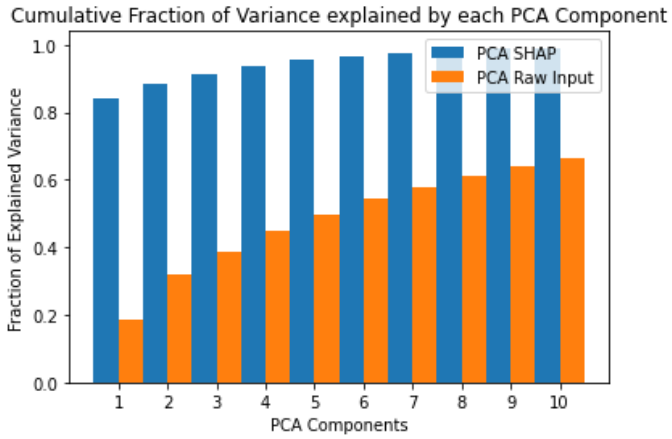


FIGURE 4.5: Cumulative variance explained by each PCA component.

TABLE 4.2: Overall performance against benchmarks.

Method	Accuracy (%)	Recall (%)	Precision (%)	XAI
Proposed Solution	93.28	97.81	91.05	✓
Yang *et al.* [6]	89.36	84.86	95.98	✗
Javaid *et al.* [17]	88.39	95.95	85.44	✗
Wang *et al.* [10]	80.6	80.6	82.8	✓
Marino *et al.* [8]	95.5	-	-	✓

Table 4.2 compares the overall performance of our pipeline against various state-of-the-art works from the literature. As seen in Table 4.2, our solution is able to outperform all but one of the considered benchmarks in terms of its overall accuracy, falling short

by only 2.2% compared to [172]. In addition, our solution is able to outperform all benchmarks in terms of its recall rate, while only slightly under-performing in terms of its precision compared to the work in [140]. While these results suggest that our solution is capable of achieving satisfactory performance compared to the state-of-the-art for binary classification, we hope to extend our analysis to the case of multiclass classification in future work.

4.5 Chapter Conclusion

In this chapter, we have presented a five-staged pipeline for robust network intrusion detection. The main parts of proposed pipeline consists of an initial front-end model based on the XGBoost algorithm to perform supervised intrusion detection, followed by a DAE model trained on SHAP explanations to perform an additional layer of anomaly detection. Importantly, by combining the detection capabilities of both these models into our overall NIDS, our evaluation results reveal a number of key insights that aid in addressing RQ1 and RQ3 as part of our contribution, C2. First, our results reveal that the ability of our overall NIDS to detect zero-day attacks is significantly enhanced by the addition of the second stage, while our overall solution is able to outperform numerous state-of-the-art works in terms of its accuracy, recall and precision on the NSL-KDD dataset. This overall finding supports our main claim in this work; that explanations coupled with additional AI/ML can be used to enhance the robustness of the system against zero-day attacks, while it also enhances the overall performance of the system. Secondly, we note since our fourth stage proposes the adoption of unsupervised learning, that these gains are effectively learnt automatically by the system, i.e., there is no need for the HITL to manually label the training data required for the fourth stage. In addition, we have also outlined two edge cases whereby the individual predictions from the second and fourth stages of our overall NIDS can be jointly compared to aid in minimising the effort of the HITL, for example, by applying an automated response to a sample deemed as malicious but not anomalous. Finally, we also consider the insights gained from the explanations of our system as another potential aid which may help reduce the effort of the HITL in a practical context, for example, by helping the HITL to understand why a particular prediction has been classified as an attack, or vice versa. These findings conclude our second

contribution, C2.

In the next chapter, we present our third contribution, C3, which aims to provide a direct answer to both our fourth research question, RQ4, as well as our first and foremost research question, RQ1.

Chapter 5

A General Methodology for Improving Network Intrusion Detection through XAI

This chapter presents our third contribution, C3, which addresses RQ4 and RQ1. In particular, we recall that RQ4 relates to the in-depth validation to verify the findings of our proposed methodology presented in Chapter 4, while RQ1 relates to the broader integration of XAI into high-stakes areas of the networking domain to enhance the overall reliability and service of the network.

In this chapter, we first extend the scope of our methodology from Chapter 4, such that our final proposed methodology supports general improvements in the overall performance of the system against any attack type, i.e., no longer limited to only zero-day attacks. Moreover, we formulate our contribution to this chapter by performing an in-depth evaluation analysis comprising multiple experiments to assess and validate the generalisation capability of our final proposed methodology under various conditions and use cases. Specifically, our analysis begins by first applying our proposed methodology across three prominent cybersecurity datasets, including the NSL-KDD, CICIDS2017 and CSE-CIC-IDS2018 datasets, and comparing the achieved performance gains against the state-of-the-art in NIDSs. The results reveal that our methodology is able to consistently outperform the considered benchmarks on common intrusion metrics (e.g., accuracy, recall, precision, F1-score, etc.), and that a large portion of the achieved gains can be directly attributed to the XAI and additional ML blocks that form part of our methodology. We then perform a further series of evaluations and experiments to assess the

robustness of our methodology under various conditions, such as i) when the type of model used by the front-end NIDS is varied, ii) when the XAI methodology is varied, and iii) we assess the ability of our methodology to generalise to unseen data distributions, i.e., training on one dataset/domain and evaluating on a different dataset/domain. The final results of these experiments indicate that our methodology can support a wide variety of AI/ML and XAI algorithms, while its ability to generalise to unseen datasets is comparable to that of the state-of-the-art.

In summary, our main contributions in this chapter include:

- We propose an extension to our previous methodology in Chapter 4 to support general improvements in the overall performance of the system against any attack type.
- Evaluation of our proposed methodology across multiple intrusion datasets, and a comparison against the state-of-the-art.
- An analysis of the robustness of our methodology under various conditions, such as varying the type of front-end model and XAI methodology used, as well as assessing the ability of our methodology to generalise to unseen environments.

To the best of our knowledge, this work presents the first to extensively assess how well the use of explanations for model improvement generalises within the cybersecurity and intrusion detection context.

The remainder of this chapter is organised as follows: in Section 5.1, we discuss the details of our extended methodology to support general improvements against any attack type. In Section 5.2, we describe the details of our experimental analysis to assess the performance and general robustness of our methodology, while in Section 5.3, we present and discuss the findings of each of our experiments. Finally, we offer our conclusions in Section 5.4.

5.1 Extended Methodology

This section presents the details of our extended methodology for enhancing intrusion detection performance through the use of additional AI/ML tools to automatically analyse explanations. In particular, we recall from Chapter 4 that our proposed methodology

comprises five main stages, including i) a data collection and cleaning stage, ii) a front-end NIDS to perform preliminary intrusion detection of incoming network traffic, iii) a feature attribution-based XAI stage to explain the behaviour of the front-end NIDS, iv) an anomaly detection stage to identify unseen explanation behaviours and v) an output update and visualisation stage. Crucially, while the original motivation of our methodology presented in Chapter 4 centered on leveraging additional AI/ML to enhance intrusion detection performance specifically against zero-day attacks, in this chapter, we are interested in extending the functionality of our methodology to support general improvements in the overall performance of the system against any attack type. Additionally, we also aim to maximise the applicability of our methodology across a broader range of implementation variations by exploring its feasibility under different choices of algorithms employed at the front-end NIDS and XAI stages of our methodology. Under these aims, we propose the following modifications and considerations at the second, third and fourth stages of our original methodology, while leaving the remaining stages unchanged:

- **Front-End NIDS**: We recall from Chapter 4 that the original implementation of the front-end NIDS was limited to a single class of commonly adopted ML algorithms for intrusion detection, namely tree-based models, such as the XGBoost classifier. In this chapter, we extend the scope of our methodology by exploring the performance trade-offs when the front-end classifier is implemented under another widely adopted class of ML algorithms for intrusion detection applications, namely a DNN-based classifier. This investigation allows us to assess the broader applicability of our methodology across diverse model architectures commonly employed in intrusion detection problems.

- **Feature Attribution (XAI)** We recall from Chapter 4 that the original implementation of our methodology was limited to a single feature attribution method, namely the "TreeExplainer" method from [81], which could provide fast and accurate SHAP-based explanations specifically for tree-based models, such as the front-end NIDS. In this work, we further extend the scope of our methodology by exploring the performance trade-offs when the third stage is implemented using a broader set of widely studied feature attribution methods from the literature, including SHAP,

LIME [59], Saabas [68], Integrated Gradients [71], and the recently proposed MAPLE [70] method. In addition, we also investigate the performance of our methodology when the third stage is composed of a different family of explanation methods - namely, the class of counterfactual XAI - by considering a state-of-the-art counterfactual-based method proposed by [173]. Overall, this broader investigation allows us to assess the impact of various explanation techniques on the overall performance of our methodology.

- **Anomaly Detection** In the previous chapter, we showed that a DAE setup similar to that shown in Fig. 5.1 can be used to reliably detect instances of new attacks within the network whenever the training distribution used to train the DAE, $\hat{\mathcal{D}}_{\phi^{(train)}}$, is specifically constructed using explanations from both the benign and previously known attack classes. While this ability to detect zero-day attacks presents a significant finding on its own, in this work, we also aim to extend the capacity of our methodology to achieve general improvements in overall system performance by detecing both new attacks and instances of known attacks that the front-end NIDS may fail to correctly identify. To achieve this extended functionality, we propose the idea of training the DAE exclusively on explanations derived from the benign class of the training data. By doing so, the resulting DAE can be explicitly optimised to assign the "normal" class to any benign traffic and the "anomaly" class to *any other* type of traffic, i.e., known attack traffic or zero-day attacks. As we later demonstrate through our experimental results in Section 5.3, this subtle yet critical adjustment to the training process of our fourth stage can lead to significant improvements in our methodology's ability to detect against any general attack types.

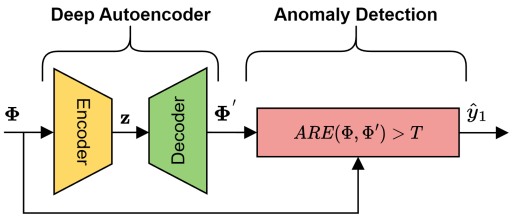


FIGURE 5.1: Detailed outline of Anomaly Detection stage.

In Figure 5.2, we summarise the overall stages of our extended methodology, emphasising the key modifications that we make in relation to our original methodology.

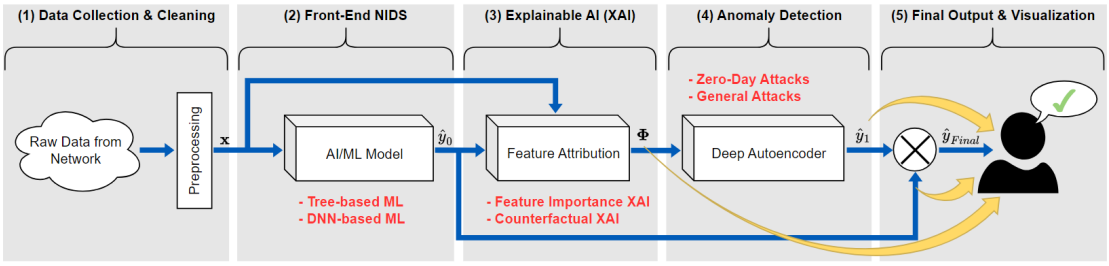


FIGURE 5.2: Overview of our extended methodology with key modifications in red.

5.2 Experimental Analysis

In this section, we present the details of our experimental analysis that we performed to evaluate the overall performance and robustness of our proposed methodology presented in Section 5.1. Specifically, in Section 5.2.1, we outline and discuss each of the experiments that were performed as part of our analysis, while in Section 5.2.2, we summarise the various datasets used in our analysis.

5.2.1 Overview

We conducted four main experiments to assess the overall performance and robustness of our proposed methodology. In the first experiment, we evaluate the performance of our methodology across multiple intrusion datasets and under multiple common intrusion metrics. In the second and third experiments, we then evaluate the performance trade-offs of our methodology when i) the front-end intrusion model is devised using a neural network-based topology, and ii) the XAI technique employed as part of the third stage of our methodology is varied according to the various XAI methodologies previously discussed in Section 4.2.3. Finally, the fourth experiment aims specifically to assess the generalisation power of our methodology to unseen environments by implementing the state-of-the-art inter-evaluation strategy proposed by [174]. Below, we elaborate further on each of these experiments:

Experiment 1: Performance Across Multiple Datasets

In this experiment, we assess the general improvement gains that can be achieved by our methodology when applied across numerous intrusion datasets (NSL-KDD, CICIDS2017 and CSE-CIC-IDS2018) and under various common intrusion metrics (accuracy, recall,

precision, F1-score). We summarise the main steps performed during this experiment in algorithm 1.

Given a particular dataset to evaluate, we first pass the dataset through the preprocessing stage, which cleans and splits the data into the relevant train and test sets (line 2). The train set is then used to optimise the parameters of the front-end NIDS (line 3), after which the initial predictions, $\hat{Y}_0$, are obtained (line 4) and a preliminary evaluation of the front-end NIDS is performed (line 5). During the third stage of our methodology, we leverage feature attribution-based XAI to explain the predictions of the front-end NIDS across the training and test samples (lines 8 and 9, respectively), noting that in the former case, we only require benign samples from the training data to be explained as this allows us to optimise our methodology against any attack class during testing. The resulting set of explanations, $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}$, then subsequently becomes the training dataset used by the DAE in the next stage. Moreover, to allow the DAE to be fairly optimised according to the specific characteristics of each dataset, we first perform a grid search to determine suitable values for the most important hyperparameters of the DAE (lines 10 and 15-23). Specifically, we consider the number of layers, number of neurons per layer, and activation function of each layer as potential hyperparameters to optimise and pass the range of values to consider during the grid search as an input variable at the start of the experiment ("grid_params"). Following its final training phase (line 11), the DAE is then used to perform anomaly detection across the test set (line 12), after which the final adjusted predictions are calculated (line 13) and the overall performance of our methodology is evaluated (line 14).

Importantly, for each repeat of the experiment across the considered datasets, we implement our methodology based on the following common parameters: For the front-end NIDS, we implement an XGBoost model using the Scikit-Learn [170] library for Python and use its default parameters for binary classification. For the third stage of our methodology, we implement the TreeExplainer methodology [81], which proved itself to be a good explanation method in the previous chapter of our work. Finally, for the threshold, T, used by our anomaly detection stage, we consider any test samples with an ARE greater than the 95th percentile of ARE occurring across the reconstructed training data to be anomalous.

In the case of the preprocessing stage, which differs slightly according to each of the

datasets, we perform the following: for the NSL-KDD dataset, we implement the same preprocessing pipeline as outlined in our previous work [175], i.e., applying label encoding to any categorical features and normalising features to the range [0,1]. In the case of the CICIDS2017 and CSE-CIC-IDS2018 datasets, we adopt the same pre-processing methodology as described in the paper [176], which involves tasks such as removing any samples containing missing or duplicate values, feature selection, and data normalisation, etc.

Algorithm 1: Evaluate Performance Across Multiple Datasets

Input : $Dataset \in$ {NSL-KDD, CICIDS2017, CSE-CIC-IDS2018}, $grid\_params$
Output: $Results_{front\_end}, Results_{overall}$

$Results_{front\_end}, Results_{overall} = \{\}$
$X_{train}, Y_{train}, X_{test}, Y_{test} \leftarrow Preprocess(Dataset)$
$Front\_End \leftarrow model.train(X_{train}, Y_{train})$
$\hat{Y}_0 \leftarrow Front\_End.predict(X_{test})$
$Results_{front\_end} \leftarrow Evaluate(\hat{Y}_0, Y_{test})$
$X_{temp} \leftarrow X_{train} \in \{benign\}$
$\hat{Y}_{temp} \leftarrow Front\_End.predict(X_{temp})$
$\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}} \leftarrow XAI.explain(X_{temp}, \hat{Y}_{temp})$
$\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}} \leftarrow XAI.explain(X_{test}, \hat{Y}_0)$
$\gamma^* = Grid\_Search(grid\_params)$
$DAE = model.train(x, y = \boldsymbol{\phi}_{train}, \gamma^*)$
$\hat{Y}_1 \leftarrow Anomaly\_Detection(dae, \boldsymbol{\phi}_{train}, \boldsymbol{\phi}_{test})$
$\hat{Y}_{Final} = max(\hat{Y}_0, \hat{Y}_1)$
$Results_{\text{overall}} \leftarrow Evaluate(\hat{Y}_{Final}, Y_{test})$;
Function *Grid_Search* $(grid\_params)$**:**
    $\gamma^* = \{\}$ // record best hyperparameters
    $\zeta^* = 999$ // record lowest loss
    for γ *in* $grid\_params$ **do**
        $DAE = model.train(x, y = \boldsymbol{\phi}_{train}, \gamma)$
        if $DAE.loss() < \zeta^*$ **then**
            $\gamma^* \leftarrow \gamma$ // update γ^*
            $\zeta^* \leftarrow DAE.loss()$ // update ζ^*
    return γ^*

Experiment 2: Varying Front-End NIDS

While our first experiment aims to assess the generalisability of our methodology to multiple datasets, its overall scope presents limitations in the sense that the type of ML model used to devise the front-end NIDS remains constant across each repetition of the experiment, i.e., we only consider an XGBoost model (i.e., tree-based ensemble ML). Therefore, to complement that experiment, experiment 2 investigates the ability of our methodology

to generalise to cases where the front-end NIDS stems from another family of ML models, namely neural networks, which are prominently employed in modern-day cybersecurity applications. In particular, we formulate part of our second experiment based on an implementation of the work presented in [177], in which the authors propose a DAE model to perform binary intrusion detection. Here, we adopt a similar model architecture as used in [177] to serve as the basis of the front-end NIDS, and then integrate the remaining stages of our methodology into the overall ML pipeline to examine the performance gains that are subsequently achieved. As indicated by Algorithm 2, we adopt a similar experimental procedure as used in experiment 1, but with the following main differences:

i) We focus our analysis solely across the NSL-KDD dataset, which is the same dataset used by the authors in [177] to develop and evaluate their proposed model.

ii) To improve the statistical confidence of our results, we apply a 10-fold cross-validation strategy across the NSL-KDD dataset, thereby allowing our final results to be based on a distribution of performance measurements rather than a single point average (i.e., indicated by lines 2, 7, and 16).

iii) Since the TreeExplainer method used during experiment 1 does not apply to neural network-based models, we instead leverage the IG method [71] to devise explanations of the front-end NIDS.

Additionally, as the main idea of this experiment is to substitute the front-end NIDS with the intrusion model proposed by [177], we also implement the same preprocessing and training steps as outlined in the author's original paper during the corresponding stages of our experiment. Finally, similar to before, we consider the 95th percentile of ARE across the training distribution, $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}$, as the threshold for detecting anomalies during the fourth stage of our methodology.

Experiment 3: Varying XAI Methodologies

As many different XAI methodologies have been proposed in the literature to date, an important implementation detail of our solution relates to the choice of XAI technique to adopt. This is especially relevant when considering the fact that many, if not all, of

Algorithm 2: Varying Front-End NIDS

Input : $Dataset \in \{\text{NSL-KDD}\}$, $grid\_params$
Output: $Results_{front\_end}$, $Results_{overall}$

$Results_{front\_end}, Results_{overall} = \{\}$
for *fold_i in {1,...,10}* **do** `// for each fold`
  $Dataset' \leftarrow Random\_Split(Dataset)$
  $X_{train}, Y_{train}, X_{test}, Y_{test} \leftarrow Preprocess(Dataset')$
  $Front\_End \leftarrow model.train(X_{train}, Y_{train})$
  $\hat{Y}_0 \leftarrow Front\_End.predict(X_{test})$
  $Results_{front\_end}\{fold\_i\} \leftarrow Evaluate(\hat{Y}_0, Y_{test})$
  $X_{temp} \leftarrow X_{train} \in \{benign\}$
  $\hat{Y}_{temp} \leftarrow Front\_End.predict(X_{temp})$
  $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}} \leftarrow XAI.explain(X_{temp}, \hat{Y}_{temp})$
  $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}} \leftarrow XAI.explain(X_{test}, \hat{Y}_0)$
  $\gamma^* = Grid\_Search(grid\_params)$
  $DAE = model.train(x, y = \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}, \gamma^*)$
  $\hat{Y}_1 \leftarrow Anomaly\_Detection(DAE, \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}, \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}})$
  $\hat{Y}_{Final} = max(\hat{Y}_0, \hat{Y}_1)$
  $Results_{\text{overall}}\{fold\_i\} \leftarrow Evaluate(\hat{Y}_{Final}, Y_{test})$

the existing XAI methodologies proposed in the literature, tend to exhibit varying trade-offs in the time required to produce a single explanation as well as the corresponding faithfulness or accuracy of the explanation to capture the true underlying behaviour of the model. Thus, the ability to produce consistent improvement gains regardless of the adopted XAI technique can have significant implications on the feasibility of our proposed methodology in real-world scenarios. Therefore, in this experiment, we aim to complement the findings of our previous two experiments by evaluating the ability of our methodology to generalise across different choices of XAI methodology.

We outline the various steps performed during this experiment in Algorithm 3. Similar to our previous experiment, we apply our methodology across the popular NSL-KDD dataset and begin by performing any necessary preprocessing tasks followed by training the front-end NIDS (for simplicity, we implement the same preprocessing steps and XGBoost model for the front-end NIDS as was used in experiment 1). We then separately implement each of the considered XAI methodologies to explain the predictions of the front-end model across the train and test sets and then apply the remaining stages of our methodology so that we can later compare the overall performance gains achieved by each of the considered XAI methodologies. Specifically, we consider four

main XAI methodologies in our assessment, including the previously examined TreeExplainer method [81], and three additional XAI methodologies not already examined in our previous experiments, including LIME [59], Saabas [69] and MAPLE [70].

Algorithm 3: Multiple XAI methodologies

Input : $Dataset \in$ {NSL-KDD}, $grid\_params$,
$XAI \in$ {TreeExplainer, LIME, Saabas MAPLE}
Output : $\text{Results}_{TreeExplainer}$, Results_{LIME}, Results_{MAPLE}

$X_{train}, Y_{train}, X_{test}, Y_{test} \leftarrow Preprocess(Dataset)$
$Front\_End \leftarrow model.train(X_{train}, Y_{train})$
$\hat{Y}_0 \leftarrow Front\_End.predict(X_{test})$
$X_{temp} \leftarrow X_{train} \in \{benign\}$
$\hat{Y}_{temp} \leftarrow Front\_End.predict(X_{temp})$
for *XAI in {TreeExplainer, LIME, MAPLE}* **do**
  $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}} \leftarrow XAI.explain(X_{temp}, \hat{Y}_{temp})$
  $\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}} \leftarrow XAI.explain(X_{test}, \hat{Y}_0)$
  $\gamma^* \leftarrow Grid\_Search(grid\_params)$
  $DAE = model.train(x, y = \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}, \gamma^*)$
  $\hat{Y}_1 \leftarrow Anomaly\_Detection(DAE, \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}, \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}})$
  $\hat{Y}_{Final} = max(\hat{Y}_0, \hat{Y}_1)$
  $\text{Results}_{\text{overall}}\{XAI\} \leftarrow Evaluate(\hat{Y}_{Final}, Y_{test})$

Experiment 4: Generalisation to Unseen Environments

Motivated by the fact that many works tend to overestimate the real-world performance of their intrusion models [178], the authors in [174] have recently proposed a novel inter-dataset evaluation strategy aimed at assessing the generalisation performance of various intrusion models. Under this strategy, one dataset is employed for training and validating the AI/ML model, while a different (but feature-wise compatible) dataset is used to evaluate the performance of the model in an unbiased manner. In this experiment, we assess the generalisation capability of our methodology by applying the inter-evaluation strategy proposed by [174] across the CICIDS2017 and the CSE-CIC-IDS2018 datasets.

We illustrate the overall steps performed during our experiment in Algorithm 4. During the first run of the experiment, we use the train set of the CICIDS2017 dataset (i.e., "Dataset_A") to train the front-end NIDS and DAE model used in our methodology and use the test set of the CSE-CIC-IDS2018 dataset (i.e., "Dataset_B") to evaluate the overall performance gains that are achieved. We then repeat our experiment using the train set of the CIC-IDS2018 dataset (i.e., "Dataset_A") to train each model and the test set of the

CICIDS2017 dataset (i.e., "Dataset_A") to evaluate the performance gains. Moreover, we implement the same preprocessing steps as outlined in [174] to clean and prepare the CICIDS2017 and the CSE-CIC-IDS2018 datasets, and train an XGBoost model to serve as the basis for the Front-end NIDS during each repetition of the experiment.

Algorithm 4: Unseen Environments (Inter-Evaluation)

Input : $Dataset\_A$, $Dataset\_B$, $grid\_params$
Output : $\text{Results}_{front\_end}$, $\text{Results}_{A->B}$

$X^A_{train}, Y^A_{train}, X^A_{test}, Y^A_{test} \leftarrow Preprocess(Dataset\_A)$
$X^B_{train}, Y^B_{train}, X^B_{test}, Y^B_{test} \leftarrow Preprocess(Dataset\_B)$
$Front\_End \leftarrow model.train(X^A_{train}, Y^A_{train})$
$\hat{Y}_0 \leftarrow Front\_End.predict(X^B_{test})$ `//apply dataset B`
$X_{temp} \leftarrow X^A_{train} \in \{benign\}$
$\hat{Y}_{temp} \leftarrow Front\_End.predict(X_{temp})$
$\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}} \leftarrow XAI.explain(X_{temp}, \hat{Y}_{temp})$
$\hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}} \leftarrow XAI.explain(X^B_{test}, \hat{Y}_0)$ `//apply dataset B`
$\gamma^* \leftarrow Grid\_Search(grid\_params)$
$DAE = model.train(x, y = \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}, \gamma^*)$
$\hat{Y}_1 \leftarrow Anomaly\_Detection(DAE, \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(train)}}, \hat{\mathcal{D}}_{\boldsymbol{\phi}^{(test)}})$
$\hat{Y}_{Final} = max(\hat{Y}_0, \hat{Y}_1)$
$\text{Results}_{A->B} \leftarrow Evaluate(\hat{Y}_{Final}, Y^B_{test})$

5.2.2 Datasets

In addition to the NSL-KDD dataset previously outlined in Chapter 4, we consider the following additional datasets in this chapter:

CICIDS2017: Released by the Canadian Institute for Cybersecurity (CIC) in 2017, the CICIDS2017 dataset [176] captures realistic network traffic data across a duration of 5 days, during which a victim network of 12 machines were attacked based on 7 common and up-to-date attack types (e.g., DoS, DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port scan and Botnet). The final dataset has been made available to the public in two main formats, including the raw PCAP files that were collected during this period and a CSV file, in which the authors have applied some initial pre-processing steps to the original PCAP files using the CICFlowMeter analyser software. In our work, we use the CSV format of this dataset, which includes over 2.8 million samples capturing both benign and attack instances, with each sample comprising over 80 different network traffic features extracted by the CICFlowMeter software as well as the corresponding class label.

CSE-CIC-IDS2018: Released one year after the CICIDS2017 dataset, the CSE-CIC-IDS2018 dataset [179] represents the result of a collaboration between the CIC and the Communications Security Establishment (CSE) to produce an updated intrusion dataset. In comparison to the CICIDS2017 dataset, this dataset shares a similar set of features and attack types but has been generated across a much larger network of victim and attack nodes. In this case, the final dataset has been generated across a victim network composed of 420 machines and 30 servers, while the attacking infrastructure consists of 50 machines. Importantly, as pointed out by [174], the combination of compatible features shared between the CICIDS2017 and CSE-CIC-IDS2018 datasets as well as having differing network setups used during the traffic data generation process, makes the use of both datasets ideal in assessing the generalisation power of a NIDS, i.e., by training the NIDS on one dataset and assessing its performance on the other. We therefore use both datasets during our initial performance evaluations as well as during our generalisation experiment.

5.3 Results

In this section, we present and discuss the findings of our experimental analysis outlined in Section 5.2. In our experiments, we consider a number of conventional intrusion detection metrics, including accuracy, recall, precision, and F1-score (we refer the reader to Section 2.1.2 for further details on each of these metrics). Additionally, we also consider the "Local Accuracy" metric from [81] to provide an indication of the accuracy of the explanations produced by each of the XAI methodologies investigated as part of our analysis (we refer the reader to Section 2.1.5 for further details on this metric).

5.3.1 Performance Across Multiple Datasets (Experiment 1)

We summarise the results of our first experiment in Tables 5.1-5.3. Here, each table corresponds to the results measured across one of the three datasets used in our experiment and includes both the preliminary performance achieved by the front-end NIDS (i.e., first row) as well as the performance achieved by our overall methodology (i.e., second row). For reference, we also include performance results from several recent intrusion detection studies on each dataset to allow a comparison of our findings against the current

state-of-the-art. Specifically, for the NSL-KDD dataset (Table 5.1), we consider the works of [140], [171], [144], and [180] as suitable benchmarks, as these studies propose recent solutions for binary classification on the NSL-KDD dataset and present their results using similar performance metrics to our own. In the case of the CIC-IDS2017 dataset, we consider the works of [174] and [180], which also provide recent binary classification results under similar metrics. Additionally, we note that both of these works report performance across a diverse set of ML algorithms — In this case, results across a total of eight distinct ML models are given — allowing us to better contextualise the significance of our results. Following similar criteria, we consider the works of [174] and [181] as suitable benchmarks for comparing our results on the CSE-CIC-IDS2018 dataset; notably, these studies report performance results across a total of nine different ML algorithms for binary classification on this dataset.

Starting with Table 5.1, which shows the performance of our methodology across the NSL-KDD dataset, a number of important insights can be drawn. First, a comparison between the performance of the front-end NIDS and our overall methodology reveals a significant gain in the performance of the latter across three of the four considered metrics; in this case, the addition of the third (i.e., XAI layer) and fourth layers (i.e., DAE layer) of our methodology have lead to gains of approximately +14% in accuracy, +32% in recall, +16% in F1-score, while the level of precision reduces by slightly under 6%. Importantly, while these gains in the former three metrics present favourable outcomes of our methodology, it is worth noting that the reduction in precision can be mainly attributed to our choice of threshold value used to distinguish anomalies from benign samples during the fourth stage of our methodology (i.e., here we used the 95th percentile of ARE across the training data as the threshold value), and can be further fine-tuned in practice to obtain a more favourable trade-off between the level of recall and precision achieved by our methodology.

Finally, a comparison against the considered benchmarks shows that our methodology achieves the highest level of accuracy (93.39%), recall (97.92%) and F1-score (94.40%), while the level of precision (91.12%) still manages to rank second highest among the various benchmark studies shown. While these results give a promising early indication of the performance gains that are achievable by our methodology, we next consider the remaining results shown in Tables 5.2 and 5.3, which correspond to the performance of our

methodology across the CIC-IDS2017 and CSE-CIC-IDS2018 datasets, respectively.

Interestingly, for both the CIC-IDS2017 and CSE-CIC-IDS2018 datasets we find that the differences in performance between the initial front-end NIDS and our overall methodology become less apparent than before; in this case, falling within $\pm 2\%$ of each other across each of the considered metrics and on both datasets. At the same time, a comparison against the various benchmarks for each dataset shows that our methodology still achieves satisfactory high levels of performance, in this case, alternating between ranking the highest and second highest on each metric across both datasets.

Importantly, while our comparison against the state-of-the-art presents satisfactory results for our methodology, we believe that a direct comparison between the performance of our front-end NIDS and overall methodology becomes increasingly challenging as the performances of both algorithms approach their maximum achievable values on each metric (i.e., as the performance gets closer to a score of 100%). To further support this claim, we demonstrate later following our results of experiment 4, that despite the initial near-perfect results obtained by the front-end NIDS during this part of our analysis, the performance of the front-end NIDS still degrades significantly when cross-evaluated on unseen data, in contrast to our overall methodology, which still manages to maintain comparable results with other state-of-the-art benchmarks.

TABLE 5.1: Performance across NSL-KDD dataset.

	Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
Proposed	XGBoost	79.20	65.61	96.82	78.22
	XGBoost-DAE	93.39	97.92	91.12	94.40
Yang *et al.* [140]	SAVAER-DNN	89.36	84.86	95.98	90.08
Javaid *et al.* [171]	Softmax Regression	88.39	95.95	85.44	90.40
Wang *et al.* [144]	One-vs-All	80.6	80.6	82.8	80.7
Sirisha *et al.* [180]	Logistic Regression	72.25	91.94	61.99	74.05
	Decision Tree	75.45	96.75	64.29	77.25
	Random Forest	76.50	97.22	65.25	78.09
	Gaussian NB	74.34	86.90	65.16	74.47
	K-Nearest Neighbors	76.41	96.26	65.36	77.85

5.3.2 Varying Front-End NIDS (Experiment 2)

We summarise the results of our second experiment in Fig. 5.3, which comprises three sub-figures, each containing box plots used to capture the distribution of results recorded across our 10-fold evaluation. Fig. 5.3a shows the preliminary performance obtained by

TABLE 5.2: Performance across CICIDS2017 dataset.

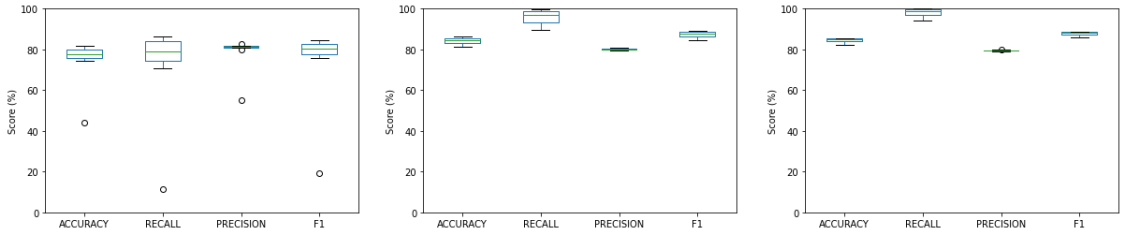
	Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
Proposed	XGBoost	98.82	98.41	99.86	99.13
	XGBoost-DAE	97.65	99.31	97.29	98.29
Verkerken *et al.* [174]	PCA	90.98	94.35	93.46	93.90
	Isolation Forest	91.11	93.14	94.70	93.91
	Autoencoder	94.26	97.78	94.59	96.16
	One-Class SVM	92.23	99.20	91.04	94.95
Sirisha *et al.* [180]	Logistic Regression	82.30	65.42	54.08	59.21
	Decision Tree	89.16	94.73	65.48	77.44
	Random Forest	93.77	84.15	84.15	84.15
	Gaussian NB	69.67	47.19	31.70	37.93
	K-Nearest Neighbors	90.69	98.41	68.23	80.59

TABLE 5.3: Performance across CSE-CIC-IDS2018 dataset.

	Algorithm	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
Proposed	XGBoost	93.51	92.41	98.59	95.40
	XGBoost-DAE	93.32	93.64	97.08	95.33
Verkerken *et al.* [174]	PCA	86.37	94.81	87.52	91.02
	Isolation Forest	87.90	91.07	92.23	91.65
	Autoencoder	87.66	91.64	91.44	91.54
	One-Class SVM	88.98	93.23	92.68	92.96
Dini *et al.* [181]	SVM	87.11	78.80	84.56	81.58
	ELM	89.01	88.88	89.45	89.16
	Poly BR	94.50	93.20	94.39	93.79
	Poly PCA	92.54	92.11	92.33	92.22

the front-end NIDS that has been implemented based on the DAE model proposed by [177], while Figs. 5.3b and 5.3c correspond to the performance achieved by our overall methodology when the third stage has been implemented using the IG and DeepExplainer XAI methodologies, respectively. For this experiment, we again consider accuracy, recall, precision, and F1 score.

Similar to our first experiment, a comparison between the performance of the front-end NIDS and our overall methodology reveals the ability of the latter to yield significantly enhanced results across the considered metrics. Specifically, in the case where IG has been implemented as part of the third stage of our methodology, we find that the average gains in performance can be summarised as +9.4% in accuracy, +22.9% in recall, +1.4% in precision and +12.9% in F1-score in comparison to those achieved by just the front-end NIDS on its own. In the case where the DeepExplainer methodology has been employed, even higher gains can be observed across most of the considered metrics, in this case equating to +9.7% in accuracy, +25.2% in recall, +0.7% in precision, and +13.4%



(A) Front-End NIDS only (B) Overall Methodology (Integrated Gradients) (C) Overall methodology (DeepExplainer)

FIGURE 5.3: Box plots showing the performance (10-fold evaluation) of our methodology when the front-end NIDS is implemented based on the DAE model from [177], and IG and DeepExplainer used by the third stage of our methodology.

in F1-score.

In addition to observing higher gains on average, a surprising result also emerges when we consider the variance or spread in performance observed across each plot. In this sense, we find that while the performance of the front-end NIDS tends to exhibit relatively high amounts of variability across the 10-fold evaluation, our methodology as shown by Figs. 5.3b and 5.3c tends to produce much more stable gains in performance. As a numerical example, when we examine just the accuracy metric alone, we find that the accuracy of the front-end NIDS can be characterised by a standard deviation of $\approx$ 11.0%, while the standard deviation of our overall methodology shown in Figs. 5.3b and 5.3c corresponds to $\approx$ 1.65% and $\approx$ 1.02%, respectively.

Perhaps even more surprisingly, we highlight the ability of our methodology to produce strong performance gains even in cases where the initial performance of the front-end NIDS can be considered extremely poor. For example, we find that even on the lowest performing fold of our experiment, i.e., when the accuracy of the front-end NIDS was found to be $\approx$ 44%, the final accuracy obtained by our overall methodology corresponds to $\approx$ 81% (using IG) and $\approx$ 84% (using DeepExplainer). In future work, we plan to further investigate the effect that the initial performance of the front-end NIDS has on the final achievable gains of our methodology.

As a final outcome of our second experiment, table 5.4 shows the average inference times taken by i) the front-end NIDS, ii) our overall methodology using IG, and iii) our overall methodology using DeepExplainer, to make an single prediction across the test set. For our experimental setup, we used an i7-8750 Intel core processor running at 2.20 GHz clock speed and 16GB of RAM, while all model training and inference operations

have been performed using the TensorFlow and Keras libraries for Python. From Table 5.4, we see that our methodology incurs a time delay of roughly 2000-3000 times that taken by the front-end NIDS to make a single prediction regarding the network's state, with DeepExplainer being approximately 1/3 faster than the IG method.

While the average inference times of our methodology are still well within the sub-second range, we believe it is important to point out that the overwhelming majority of these delays stem directly from the time required by both XAI methodologies to explain the predictions of the front-end NIDS. Thus, from a practical perspective, we believe this poses a greater challenge to the wider research community to develop more efficient solutions for addressing the interpretability crisis that surrounds the use of black-box solutions. This becomes especially relevant when considering the fact that explanations are expected to become a legal requirement in the near future for any networks services that rely on ML and AI [21]. Finally, we emphasise to the reader that the primary purpose of this subsection is to provide an initial indication of the time complexity associated with our methodology, particularly in the case when the front-end NIDS is based on a neural network topology. In the next subsection, we provide additional and complementary results that highlight the associated time complexities of our solution across numerous other XAI methodologies and for the case when the front-end NIDS is based on an ensemble tree-based topology.

TABLE 5.4: Average inference times observed during Experiment 2.

	Front-End Model	**Overall Methodology (IG)**	**Overall Methodology (DeepExplainer)**
Time (s)	2.25×10^{-6}	7.24×10^{-3}	4.96×10^{-3}

5.3.3 Varying XAI Methodologies (Experiment 3)

We illustrate the results of our third experiment in Table 5.5. Here, each row corresponds to the overall performance of our methodology when the third stage has been implemented based on a different XAI methodology, as indicated by the first column. For this experiment, in addition to considering metrics, such as accuracy, recall, precision, and F1-score in our analysis, we also consider the local accuracy metric from [81] as a measure of the quality of each XAI methodology, as well as show the average time taken by each XAI methodology to produce a single explanation across the test set. We also note that in the

case of the LIME methodology, we provide two sets of results depending on whether we apply the 'discretization' hyperparameter that maps all numerical features into discrete bins before explaining each prediction, or whether each feature has been kept in its continuous form. Finally, we also report a set of results for when the XAI methodology has been implemented using a state-of-the-art counterfactual-based XAI methodology [173].

As shown by Table 5.5, the Sabaas and TreeExplainer methodologies achieve the highest and second-highest accuracy among all the considered methodologies in our analysis, respectively. Moreover, while TreeExplainer achieves the highest recall rate (97.92%), Sabaas achieves the highest level of precision (94.71%) and F1-Score (95.50%). In the case of the remaining methodologies, both variations of LIME and MAPLE achieve accuracy in the range of 83.23% – 84.75%, with the continuous variation of LIME achieving slightly higher accuracy, recall and F1-Score compared to its discretised counterpart. Finally, we see that the counterfactual RL approach produces the lowest performance in terms of accuracy (80.46%), recall (70.61%) and F1-Score (80.50%) among all the explanation methods considered; in an effort to explain this result, we consider one of the key differences between feature attribution-based and counterfactual-based explanations. Namely, while feature attribution-based methods tend to assign rankings of importance to each feature of the model, counterfactual explanations are typically designed to respect feature dependence while providing only the minimal amount of information about the features required to change the classification output of the model. In this sense, we believe that part of the reason why feature-attribution methods may be better suited to producing higher gains in comparison to counterfactual explanations, may be due to the generally higher amount of information expressed in attribution-based explanations, in contrast to those produced by counterfactual techniques. In future work, we hope to investigate this conjecture further and better understand the potential benefits and/or limitations of applying counterfactual-based XAI to our methodology.

As a final observation, we compare the local accuracy and time complexities of each XAI methodology. In terms of accuracy, we find that TreeExplainer and Saabas both achieve local accuracy scores of 1.0, implying both these methods produce explanations of high accuracy. On the other hand, both MAPLE and LIME (continuous) achieve a score of 0.0, indicating that the explanations produced by these methods are highly inaccurate. Surprisingly, we see that the discretised version of LIME achieves a score of

0.2, which suggests that the explanations produced by this method are at least somewhat accurate in decomposing the output of the model being explained. Finally, a comparison of the average time required to produce a single explanation reveals that the Saabas and counterfactual RL techniques achieve the fastest and second fastest rates, respectively; in this case, the Sabaas method is capable of producing explanations at a rate of almost 860 times faster than the TreeExplainer approach, which ranks third fastest among all of the considered methodologies. Finally, we see that the MAPLE and LIME (discretised) approaches require approximately 0.1-1 seconds to produce a single explanation, respectively, thereby limiting their potential usage to real-time intrusion detection applications.

Overall, the results indicated by Table 5.5 suggest that the Sabaas methodology offers the most advantages in terms of its ability to yield high-performance gains, accurate explanations as well as significantly low time overhead. While the counterfactual RL approach also produces explanations at a relatively low time delay, it suffers from low performance gain. In contrast, the TreeExplainer method is capable of achieving comparably strong performance gains compared to the Sabaas methodology, while still presenting the third lowest time complexity among the considered approaches. In our next and final experiment, we compare the overall ability of our methodology to generalise to previously unseen environments.

TABLE 5.5: Performance across multiple XAI methodologies.

XAI Methodology	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)	Local Accuracy	Time / Explanation (s)
TreeExplainer	93.39	97.92	91.12	94.40	1.0	0.003390
Saabas	94.85	96.33	94.71	95.50	1.0	3.95×10^{-6}
LIME (Discretized)	83.23	77.85	91.42	84.10	0.2	1.009465
LIME (Continuous)	84.75	81.32	90.93	85.90	0.0	0.008089
MAPLE	84.86	79.90	92.48	85.70	0.0	0.12835
Counterfactual RL	80.46	70.61	93.46	80.50	N/A	1.92×10^{-5}

5.3.4 Experiment 4: Generalisation to Unseen Environments

We summarise the results of our fourth and final experiment in Table 5.6. Here, the first two rows of Table 5.6 show the generalisation performance achieved by the front-end NIDS (XGBoost model) across the CICIDS2017 and CSE-CIC-IDS2018 datasets, while rows 3 and 4 show the generalisation performance achieved by our overall methodology across the CICIDS2017 and CSE-CIC-IDS2018 datasets. For each row of results, we use the "Train Set" column to signify which of the two datasets the associated algorithm

was trained on, and the "Test Set" column to indicate which dataset the algorithm was subsequently evaluated on. Similar to our previous experiments, we consider metrics such as accuracy, recall, precision and F1-score in our analysis. For reference, we also compare our results with a series of generalisation performances reported by the original authors in [174], who proposed the inter-evaluation strategy on which our experiment is based (see rows 5-12). Specifically, we note that the authors in [174] have evaluated the generalisation capabilities of four prominent ML algorithms across the CIC-IDS2017 and CSE-CIC-IDS2018 datasets using similar performance metrics to ours. By including these results, we can better contextualise the significance of our findings relative to these models.

From Table 5.6, a comparison between the front-end NIDS and our overall methodology reveals a significant enhancement in the generalisation performance by the latter. For example, in the case when both algorithms were trained on the CICIDS2017 dataset and evaluated on the CSE-CIC-IDS2018 dataset (i.e., indicated by rows 1 and 3), the accuracy of the front-end NIDS generalises to 39.78%, while the accuracy of our overall methodology generalises to 75.77% (i.e., approximately +36% gain). Similarly, when trained on the CSE-CIC-IDS2018 and evaluated on the CICIDS2017, the front-end NIDS achieves an accuracy of 54.35%, while our overall methodology maintains an accuracy of 79.23% (i.e., approximately +25% gain). In general, we see that this trend repeats itself across the metrics for recall and F1-score, with our overall methodology achieving significantly enhanced performance in comparison to the front-end NIDS. Interestingly, in the case of the precision metric, we find that on both datasets, the front-end NIDS achieves a higher level of precision compared to our overall methodology. However, following a closer inspection of both datasets, we believe this result most likely stems as a systematic by-product of both datasets suffering from high-class imbalance. In particular, as the composition of benign samples in these two datasets comprise between 80%-83% of the total number of samples, we believe it is likely that the front-end NIDS becomes biased towards classifying samples as benign, which due to the high-class imbalance, allows it to achieve a high level of precision even when its overall accuracy and recall rates remain low. We note that the issue of high-class imbalance on the CICIDS2017 and CSE-CIC-IDS2018 has also been observed in some other recent intrusion detection studies [182], [183].

In comparison to the benchmark algorithms, when trained on the CICIDS2017 dataset

and assessed on the CSE-CIC-IDS2018 dataset, we find that our proposed methodology achieves the highest generalisation accuracy (i.e., 0.54% higher than the second-highest value), the third highest recall (i.e., 3.4% short of the highest value), the highest level of precision (i.e., 2.17% higher than the second-highest value), and the second highest F1-score (i.e., falling short by only 0.2% compared to the highest value). In the reverse case, i.e., when trained on the CSE-CIC-IDS2018 dataset and assessed on the CICIDS2017 dataset, we find that our proposed methodology again manages to achieve the highest level of accuracy (i.e., 3.82% higher than the second highest accuracy), the second highest recall rate (i.e., only 0.11% lower than the highest value), the highest level of precision (i.e., 1.16% higher than the next best value), and the highest overall F1-score (i.e., 1.71% higher than the second highest score).

TABLE 5.6: Generalization across CICIDS2017 & CSE-CIC-IDS2018.

	Algorithm	Train Set	Test Set	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
Proposed	XGBoost	2017	2018	39.78	18.87	92.50	31.35
		2018	2017	54.35	38.75	86.89	53.60
	XGBoost-DAE	2017	2018	75.77	89.65	79.65	84.35
		2018	2017	79.23	99.82	78.78	87.33
Verkerken *et al.* [174]	PCA	2017	2018	67.48	77.80	77.62	77.71
		2018	2017	75.41	99.51	75.13	85.62
	Isolation Forest	2017	2018	44.79	35.62	75.73	48.45
		2018	2017	63.43	82.18	72.04	76.78
	Autoencoder	2017	2018	70.97	90.25	74.99	81.91
		2018	2017	63.60	83.07	71.84	77.05
	One-Class SVM	2017	2018	75.23	93.05	77.48	84.55
		2018	2017	73.63	99.93	73.63	84.79

5.4 Chapter Conclusion

In this chapter, we have presented a general methodology, whereby explanations are leveraged alongside additional unsupervised ML to automatically enhance the performance of an intrusion system for cybersecurity. As part of our third contribution to this thesis, C3, we have performed an in-depth evaluation to assess and characterise the performance gains that can be achieved by our proposed methodology under various cases and conditions. Namely, our analysis consisted of four main experiments: in the first experiment, we examined the performance gains of our methodology when applied across multiple intrusion datasets. The overall results of this first experiment revealed that i) the

addition of the third (i.e., XAI layer) and fourth layers (i.e., DAE layer) of our methodology led to significant performance gains across the considered datasets, and in particular, in the case of the NSL-KDD dataset, while ii) our methodology achieved the highest levels of accuracy, recall, and F1-score on NSL-KDD dataset, while yielding alternating performances representing the highest and second highest on each metric across the CIC-IDS2017 and CSE-CIC-IDS2018 datasets. In the second experiment, we varied the model used by front-end NIDS and found that our overall methodology still achieved significantly higher performance in comparison to the performance of the front-end NIDS when used in isolation. In our third experiment, we then compared the performance gains of our methodology when the XAI layer was varied. Our overall results for this experiment found that the Sabaas and TreeExplainer methodologies achieved the highest and second-highest performances among the considered XAI methodologies, while both variations of LIME and MAPLE still produced noticeable performance gains and the counterfactual RL approach yielded the weakest gains among the considered XAI methodologies in our analysis. Finally, in our fourth experiment, we examined the generalisation power of our methodology when assessed on unseen datasets. From the results of this experiment, it was evident that our methodology significantly improves the ability of the NIDS to generalise to unseen datasets. For example, when trained on the CICIDS2017 dataset and subsequently evaluated on the CSE-CIC-IDS2018 dataset, the accuracy of the front-end NIDS was found to generalise to 39.78%, while the accuracy of our overall methodology was found to generalise to 75.77% (i.e., approximately +36% gain). This result was also further found to be the highest generalisation accuracy among the considered benchmarks. Overall, the findings of this chapter provide strong support for the performance gains that can achieved by our proposed methodology (RQ4), with these same gains providing further insights into how XAI can be used to enhance the performance of the network (RQ1).

Chapter 6

Conclusion and Future Directions

This thesis has explored some of the potential use cases and benefits of applying XAI within the telecommunication domain. In this final chapter, we summarise the main achievements and insights that have come to light over the course of this thesis (Section 6.1). We also discuss some of the future research directions of our research and highlight some of the future trends of XAI within the telecommunication domain (Section 6.2).

6.1 Thesis Summary & Main Achievements

The work in this thesis has been primarily motivated by the challenges associated with the safe, reliable and trustworthy deployment of black-box AI/ML solutions for current and future telecommunication networks. From this perspective, the primary research question (RQ1) that this thesis has aimed to address is "How can explainable artificial intelligence (XAI) be integrated into high-stake areas of the telecommunication domain to ensure better reliability and robust service within the network?". To address this overarching question, we have broken it down into three additional and complementary research questions. Specifically, our second research question (RQ2) relates to the adoption of quantitative XAI metrics with a proposed ground-truth to evaluate the faithfulness of the explanations and ensure that our proposed solutions remain trustworthy and actionable in real world environments. Our third research question (RQ3) relates to minimising the effort of the human-in-the-loop (HITL) by proposing a novel methodology incorporating additional ML to automatically analyse explanations and improve the performance of an intrusion detection system. Finally, our fourth research question (RQ4) concerns the comprehensive verification of the performance gains achieved by our proposed methodology in addressing RQ3.

The first contributory chapter of this thesis (Chapter 3) has addressed RQ2 and made a partial contribution towards RQ1. In this chapter, we looked at the problem of resource management, and in particular, resource reservation within sliced networks. While this particular problem has been extensively examined from the perspective of the MNO, little work has previously been done to address the challenges faced from the perspective of the slice tenant, or Service Provider (SP). Thus, as part of our first contribution, we first developed two deep learning solutions for resource reservation from the tenant's perspective and then further demonstrated the potential benefits that deep learning solutions can bring over conventional modelling approaches (e.g., ARIMA), such as improved performance at multi-step ahead reservation. Following this, we then augmented our deep-learning solutions by proposing an XAI framework to enhance the transparency of AI/ML solutions employed in resource management problems. Specifically, through the use of our own ML solution for STRR, we demonstrated how our proposed framework can aid operators to monitor, understand and debug the behaviour of their AI/ML solutions. We then formulated a procedure for quantitatively assessing the accuracy and faithfulness of the explanations in our framework against a ground-truth baseline — an aspect which has been overlooked in the existing literature (RQ2). This, in turn, fosters greater trust and reliability into the overall management of the network ecosystem while ensuring that any decisions made by the AI/ML solution remain actionable in a real-world environment.

The second contributory chapter of this thesis (Chapter 4) has addressed RQ3 and made a partial contribution towards RQ1. In this work, we turned to the high-stakes area of cybersecurity, and in particular, network intrusion detection. Here, the main objectives were to determine how XAI could be leveraged to minimise the workload of the HITL while further enhancing the overall performance of the system. Specifically, while previous works on this topic have mainly looked at the benefits of utilising explanations to improve performance through the process of feature selection [149] or model debugging [147] (including our own work showcased in Chapter 3), the vast quantity of real-time decisions occurring within the intrusion setting coupled with the stringent requirement for robustness against attacks necessitates the need for a more automated approach that can reduce, as far as possible, the tasks of HITL. To this end, we proposed and demonstrated a novel methodology whereby explanations of an intrusion system are passed as inputs

into a secondary AI/ML stage for automated processing (RQ3). Our initial proposed solution demonstrated a remarkable ability to detect zero-day attacks on the NSL-KDD dataset. However, a more thorough and convincing validation of these findings was still necessary.

The third and final contributory chapter of this thesis (Chapter 5) encompassed our proposal to address RQ4, and partially RQ1, by thoroughly validating the findings of our initial work presented in Chapter 4. In this chapter, we first extended our proposed methodology so that it could achieve performance gains in any general attack class. To assess and validate the performance gains of our methodology, we then devised a series of targeted experiments aimed at evaluating the performance of our methodology i) when applied across multiple datasets, ii) when the front-end NIDS and iii) XAI stages of our methodology are varied, and iv) when our methodology is applied to unseen data distributions. Throughout each of these experiments, we found that the XAI and additional ML stages of our methodology led to significant performance gains in comparison to the standalone front-end NIDS, while our overall methodology consistently achieved between the highest and second-highest evaluation scores compared to other state-of-the-art literature in NIDSs.

6.2 Future Research Directions

While our findings throughout this thesis have led to a progressive number of insights to be discovered, so have an equal number of challenges and gaps been uncovered that require further attention. Below, we highlight just some of the main challenges and gaps uncovered by our research that we consider to be potential directions for future research:

6.2.1 XAI Evaluation Metrics

One of the first challenges encountered during this thesis relates to the complexity required to compute the ground truth scores for the masked-based XAI metrics presented in Chapter 3. Specifically, we recall from our complexity analysis in Section 3.6.2, that the complexity for computing the ground truths of these metrics is bounded by $\mathcal{O}(NN!)$, where N refers to the number of features used by the AI/ML model. In most practical settings, this relatively large complexity may present limitations to the total number of

features that can be used by the AI/ML model, especially if the ground truths are required to be calculated in an online manner (i.e., limited to a few tens of features). To this end, one potential direction for future research could be the development of suitable alternatives for evaluating the faithfulness of the explanations under less complexity. For example, in the work of [77], the authors have recently proposed an approach that makes use of statistical randomisation techniques to test and evaluate the trustworthiness of explanation methods, such as saliency maps, which are commonly employed in image processing applications. Essentially, their proposed approach entails comparing the output explanation of a saliency method applied to a trained AI/ML model with the output explanation of the same saliency method but applied to a randomly initialised and untrained AI/ML model with a similar architecture to the first. Based on the intuition that the saliency maps of both models should be substantially different if the saliency method has truly learnt relevant aspects of the first model and similar if it hasn't, the authors' proposed approach offers, at least to some extent, a computationally friendly solution for performing a "sanity check" on the quality of the explanation method that is considered.

In the case of our own evaluation strategy presented in Chapter 3 of this thesis, one potential solution to support the calculation of ground truths when the number of input features to the model is considerably large, could be through the idea of *feature grouping*. Specifically, we consider the case where the input features to the model may be naturally grouped together, for instance, if the features consist of time-lagged variations of the same primary features. In our own STRR model from Chapter 3, this could include lagged features capturing a historical window for the tenants past traffic demand, best-effort traffic, delivered traffic, etc. In this sense, our previously presented ground-truth strategy from Chapter 3 could potentially be adapted, such that it instead permutes across every combination of grouped-feature rankings and sign, i.e., rather than the full set of possible permutations for when the features are treated individually. While this proposed approach is unlikely to be able to assess the faithfulness of an explanation to capture intra-group dependencies, we postulate that it may still be useful in assessing insights at an inter-group level.

6.2.2 Multi-Classification XAI Methodology

Another potential direction for future research is to extend our final proposed methodology from Chapter 5 to support multi-classification. For this purpose, we envision one potential approach to solving this task as being based on the *one-versus-all* concept. Essentially, for a general classification setting, this approach involves splitting a multi-class problem into multiple binary sub-tasks and training a separate binary classifier to distinguish between a common class (e.g., benign traffic) and each of the remaining classes from the full set of classes considered (e.g., probe attack, brute-force attack, denial-of-service attack, etc.). Performing multi-classification can then be achieved by selecting the class with the highest joint probability amongst all the classifiers [184].

In the case of our proposed methodology from Chapter 5, the one-versus-all concept may be applied at either the front-end NIDS stage (i.e., train multiple binary front-end NIDSs), the anomaly detection stage (i.e., train multiple anomaly detectors), or a combination of both. While the optimal choice from these three options requires further research to be determined, we postulate that applying this concept to the anomaly detector stage, either on its own or in combination with the front-end NIDS, provides a number of useful advantages. For example, in the case where a NIDS is required to be robust against detecting and identifying zero-day attacks on the network, a separate anomaly detector may be integrated into an already-existing solution by training the associated DAE model on explanations that capture the behaviour of all the already-known classes. Detecting zero-day attacks may then be possible by identifying whenever the ARE of its associated DAE is highest among the ARE of all the remaining DAEs in the overall NIDS.

6.2.3 Explainable Reinforcement Learning

While the thesis has primarily considered conventional techniques for AI/ML, such as supervised/ unsupervised AI/ML, another equally promising and relevant area of research is the application of XAI to RL, which is sometimes also referred to as Explainable Reinforcement Learning (XRL) [185]. Importantly, we believe that XAI could have a profound impact in this context, especially given how widespread the adoption of RL, and

in particular DRL, has become in the telecommunication sector in recent years, as highlighted in the survey by [186]. Moreover, we note that while XAI research into conventional AI/ML models, such as DNNs [102] or tree-based models [103], etc., have already begun to make noticeable progress in the context of telecommunications, the same cannot yet be said regarding research into explainability for RL-based models within the networking domain, which is still well-within its infancy stage.

In the context of RL, we believe that XAI can provide several important insights, in addition to those which XAI already offers to most other types of conventional AI/ML models. For example, in the work of [187], it has previously been envisioned that explanations could be used by AI experts to formulate better reward functions during the design and training of their RL agent. This, in turn, could lead to faster convergence times and encourage the agent to learn more reliable policies.

Finally, one other key aspect where we believe XAI could be of benefit in RL-based applications stems from the lack of ground truth available to evaluate the RL agent. Specifically, unlike supervised learning methods, whereby one can evaluate the performance of the trained model against a known ground truth, RL problems are generally only applied to complex problems where there is no obvious or known optimal solution in the first place. In safety-critical and other high-stakes areas of the networking domain, this constitutes a key challenge for adopting RL, as it means there is no way to prove that the agent has converged to the optimal solution. While it is unlikely that XAI, as a standalone tool, could be used to prove that the agent has learnt the optimal policy, we believe that the ability to gain insights into the agent's generalised behaviour can at least provide an initial level of trust to stakeholders that the agent has learnt to make intuitive decisions.

We believe in the near future, that the study of XRL will become a major hot topic of research within the telecommunications domain as well as in other related fields of STEM research for these same reasons.

Bibliography

[1] F. Khan, Z. Pi, and S. Rajagopal, "Millimeter-wave mobile broadband with large scale spatial processing for 5g mobile communication," in *2012 50th annual allerton conference on communication, control, and computing (Allerton)*, IEEE, 2012, pp. 1517–1523.

[2] M. Latva-Aho, K. Leppänen, *et al.*, "Key drivers and research challenges for 6g ubiquitous wireless intelligence," Tech. Rep., 2019.

[3] M. Katz, M. Matinmikko-Blue, and M. Latva-Aho, "6genesis flagship program: Building the bridges towards 6g-enabled wireless smart society and ecosystem," in *2018 IEEE 10th Latin-American Conference on Communications (LATINCOM)*, IEEE, 2018, pp. 1–9.

[4] W. Saad, M. Bennis, and M. Chen, "A vision of 6g wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2019.

[5] M. E. Morocho-Cayamcela, H. Lee, and W. Lim, "Machine learning for 5g/b5g mobile and wireless communications: Potential, limitations, and future directions," *IEEE Access*, vol. 7, pp. 137 184–137 206, 2019.

[6] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[7] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6g," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 96–103, 2020.

[8] K. Mehmood, K. Kralevska, and D. Palma, "Intent-driven autonomous network and service management in future cellular networks: A structured literature review," *Computer Networks*, vol. 220, p. 109 477, 2023.

[9] R. Qureshi, M. Irfan, T. M. Gondal, *et al.*, "Ai in drug discovery and its clinical relevance," *Heliyon*, 2023.

[10] T. Davenport and R. Kalakota, "The potential for artificial intelligence in healthcare," *Future Healthcare Journal*, vol. 6, no. 2, p. 94, 2019.

[11] G. Novakovsky, N. Dexter, M. W. Libbrecht, W. W. Wasserman, and S. Mostafavi, "Obtaining genetics insights from deep learning via explainable artificial intelligence," *Nature Reviews Genetics*, vol. 24, no. 2, pp. 125–137, 2023.

[12] J. Schmidhuber, U Meier, and D Ciresan, "Multi-column deep neural networks for image classification," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 2012, pp. 3642–3649.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[14] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64–77, 2020.

[15] D. Silver, J. Schrittwieser, K. Simonyan, *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[16] R. Britto, T. Murphy, M. Iovene, L. Jonsson, M. Erol-Kantarci, and B. Kovács, "Telecom ai native systems in the age of generative ai–an engineering perspective," *arXiv preprint arXiv:2310.11770*, 2023.

[17] A. Solon, *Telecommunications Generative AI Study*, 2023. [Online]. Available: `https://pages.awscloud.com/rs/112-TZM-766/images/Altman%20Solon_AWS_Telecoms%20Generative%20AI%20Study.pdf` (visited on 01/01/2024).

[18] Ericsson, "Employing AI techniques to enhance returns on 5G network investments," Tech. Rep., 2018. [Online]. Available: `https://www.ericsson.com/en/network-services/ai-5g-networks` (visited on 12/01/2023).

[19] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. C. Zhang, "Artificial intelligence-enabled cellular networks: A critical path to beyond-5g and 6g," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 212–217, 2020.

[20] A. Kaloxylos, A. Gavras, D. Camps Mur, M. Ghoraishi, and H. Hrasnica, "AI and ML – Enablers for Beyond 5G Networks," Tech. Rep., 2021. [Online]. Available: `https://doi.org/10.5281/zenodo.4299895` (visited on 12/01/2023).

[21] W. Guo, "Explainable artificial intelligence for 6g: Improving trust between human and machine," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39–45, 2020.

[22] 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.

[23] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2018, pp. 0210–0215.

[24] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, 2018, pp. 80–89.

[25] E. Shortliffe, *Computer-based medical consultations: MYCIN*. Elsevier, 1976, vol. 2.

[26] W. R. Swartout, "Xplain: A system for creating and explaining expert consulting programs," *Artificial intelligence*, vol. 21, no. 3, pp. 285–325, 1983.

[27] D. L. McGuinness and A. Borgida, "Explaining subsumption in description logics," *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 3, no. 00, 1995.

[28] M. Van Lent, W. Fisher, and M. Mancuso, "An explainable artificial intelligence system for small-unit tactical behavior," in *Proceedings of the National Conference on Artificial Intelligence*, Citeseer, 2004, pp. 900–907.

[29] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[30] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[31] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (xai): A survey," *arXiv preprint arXiv:2006.11371*, 2020.

[32] European Commission, *Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL laying down harmonized rules on artificial intelligence (Artificial Intelligence Act)*, 2023. [Online]. Available: https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence.

[33] ETSI, "Experiential Networked Intelligence (ENI); ENI Definition of Categories for AI Application to Networks," ETSI Group Report ETSI GR ENI 007 V1.1.1. [Online]. Available: {https://www.etsi.org/deliver/etsi_gr/ENI/001_099/007/01.01.01_60/gr_ENI007v010101p.pdf}.

[34] ETSI, "Experiential Networked Intelligence (ENI); System Architecture," ETSI Group Report ETSI GR ENI 005 V2.1.1. [Online]. Available: {https://www.etsi.org/deliver/etsi_gs/ENI/001_099/005/02.01.01_60/gs_ENI005v020101p.pdf}.

[35] E. Commission, *Proposal for a regulation laying down harmonised rules on artificial intelligence*. [Online]. Available: {https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence}.

[36] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Pearson, 2016.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

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

[39] Y. Bengio and O. Delalleau, "On the expressive power of deep architectures," in *International conference on algorithmic learning theory*, Springer, 2011, pp. 18–36.

[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, Springer, 2014, pp. 818–833.

[41] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, Elsevier, 1992, pp. 65–93.

[42] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: A survey," *Artificial Intelligence Review*, vol. 57, no. 2, p. 28, 2024.

[43] C. O. Retzlaff, A. Angerschmid, A. Saranti, *et al.*, "Post-hoc vs ante-hoc explanations: Xai design guidelines for data scientists," *Cognitive Systems Research*, vol. 86, p. 101 243, 2024.

[44] B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," *Machine Learning*, vol. 102, pp. 349–391, 2016.

[45] A. Andrzejak, F. Langner, and S. Zabala, "Interpretable models from distributed data via merging of decision trees," in *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, IEEE, 2013, pp. 1–9.

[46] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 101–115, 2020.

[47] B. Mihaljević, C. Bielza, and P. Larrañaga, "Bayesian networks for interpretable machine learning and optimization," *Neurocomputing*, vol. 456, pp. 648–665, 2021.

[48] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[49] A. H. Sung, "Ranking importance of input parameters of neural networks," *Expert systems with Applications*, vol. 15, no. 3-4, pp. 405–411, 1998.

[50] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *The Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010.

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

[52] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[53] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowledge and Information Systems*, vol. 41, no. 3, pp. 647–665, 2014.

[54] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[55] 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.

[56] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[57] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *Plos One*, vol. 10, no. 7, e0130140, 2015.

[58] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[59] 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.

[60] 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, 2018.

[61] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[62] S. Sarkar, T. Weyde, A. d. 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*, CEUR Workshop Proceedings, vol. 1773, 2016.

[63] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," *Statistical science*, vol. 16, no. 3, pp. 199–231, 2001.

[64] Y.-L. Chou, C. Moreira, P. Bruza, C. Ouyang, and J. Jorge, "Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications," *Information Fusion*, vol. 81, pp. 59–83, 2022.

[65] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," in *International Conference on Learning Representations*, 2018.

[66] S. M. Lundberg, B. Nair, M. S. Vavilala, *et al.*, "Explainable machine learning predictions to help anesthesiologists prevent hypoxemia during surgery," *bioRxiv*, p. 206 540, 2017.

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

[68] A. Saabas, *Interpreting random forests*, 2014. [Online]. Available: `http://blog.datadive.net/interpreting-random-forests` (visited on 12/08/2022).

[69] A. Saabas, *Treeinterpreter*, 2019. [Online]. Available: `https://github.com/andosa/treeinterpreter` (visited on 12/08/2022).

[70] G. Plumb, D. Molitor, and A. S. Talwalkar, "Model agnostic supervised local explanations," *Advances in Neural Information Processing systems*, vol. 31, 2018.

[71] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*, PMLR, 2017, pp. 3319–3328.

[72] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International Conference on Machine Learning*, PMLR, 2017, pp. 3145–3153.

[73] M. J. Pazzani, "Representation of electronic mail filtering profiles: A user study," in *Proceedings of the 5th international conference on Intelligent user interfaces*, 2000, pp. 202–206.

[74] B. A. Myers, D. A. Weitzman, A. J. Ko, and D. H. Chau, "Answering why and why not questions in user interfaces," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, pp. 397–406.

[75] S. Stumpf, V. Rajaram, L. Li, *et al.*, "Toward harnessing user feedback for machine learning," in *Proceedings of the 12th international conference on Intelligent user interfaces*, 2007, pp. 82–91.

[76] W. W. Cohen *et al.*, "Learning rules that classify e-mail," in *AAAI spring symposium on machine learning in information access*, Stanford, CA, vol. 18, 1996, p. 25.

[77] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, vol. 31, 2018.

[78] W. Samek, A. Binder, *et al.*, "Evaluating the visualization of what a deep neural network has learned," *IEEE Trans. Neur. Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, 2016.

[79] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[80] H. Chen, J. D. Janizek, S. Lundberg, and S.-I. Lee, "True to the model or true to the data?" *arXiv preprint arXiv:2006.16234*, 2020.

[81] S. M. Lundberg, G. Erion, *et al.*, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[82] S. K. Jagatheesaperumal, Q.-V. Pham, R. Ruby, Z. Yang, C. Xu, and Z. Zhang, "Explainable ai over the internet of things (iot): Overview, state-of-the-art and future directions," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2106–2136, 2022.

[83] J. Zhou, F. Chen, and A. Holzinger, "Towards explainability for ai fairness," in *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, Springer, 2020, pp. 375–386.

[84] V. Dignum, "The myth of complete ai-fairness," in *Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15–18, 2021, Proceedings*, Springer, 2021, pp. 3–8.

[85] T. Begley, T. Schwedes, C. Frye, and I. Feige, "Explainability for fair machine learning," *arXiv preprint arXiv:2010.07389*, 2020.

[86] S. Chakraborty, R. Tomsett, R. Raghavendra, *et al.*, "Interpretability of deep learning models: A survey of results," in *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBD*
IEEE, 2017, pp. 1–6.

[87] C. Li, W. Guo, S. C. Sun, S. Al-Rubaye, and A. Tsourdos, "Trustworthy deep learning in 6g-enabled mass autonomy: From concept to quality-of-trust key performance indicators," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 112–121, 2020.

[88] N. Capuano, G. Fenza, V. Loia, and C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," *IEEE Access*, vol. 10, pp. 93 575–93 600, 2022.

[89] S. Wang, H. Yu, Y. Yuan, G. Liu, and Z. Fei, "Ai-enhanced constellation design for noma system: A model driven method," *China Communications*, vol. 17, no. 11, pp. 100–110, 2020.

[90] L. Pellaco, V. Saxena, M. Bengtsson, and J. Jaldén, "Wireless link adaptation with outdated csi—a hybrid data-driven and model-based approach," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, IEEE, 2020, pp. 1–5.

[91] Y. Lu, P. Cheng, Z. Chen, W. H. Mow, Y. Li, and B. Vucetic, "Deep multi-task learning for cooperative noma: System design and principles," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 61–78, 2020.

[92] C. Liaskos, S. Nie, A. Tsioliaridou, A. Pitsillides, S. Ioannidis, and I. Akyildiz, "End-to-end wireless path deployment with intelligent surfaces using interpretable neural networks," *IEEE Transactions on Communications*, vol. 68, no. 11, pp. 6792–6806, 2020.

[93] S. C. Sun and W. Guo, "Approximate symbolic explanation for neural network enabled water-filling power allocation," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, IEEE, 2020, pp. 1–4.

[94] R. Beals and J. Szmigielski, "Meijer g-functions: A gentle introduction," *Notices of the AMS*, vol. 60, no. 7, pp. 866–872, 2013.

[95] T. Pulkkinen, J. K. Nurminen, and P. Nurmi, "Understanding wifi cross-technology interference detection in the real world," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2020, pp. 954–964.

[96] J. Chen, S. Miao, H. Zheng, and S. Zheng, "Feature explainable deep classification for signal modulation recognition," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2020, pp. 3543–3548.

[97] A. U. Chaudhry and H. R. Hafez, "On finding hidden relationship among variables in wifi using machine learning," in *2020 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2020, pp. 157–161.

[98] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2398–2412, 2020.

[99] C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *Optical fiber communication conference*, Optica Publishing Group, 2018, W3A–4.

[100] C. Lu, D. Liang, D. Wang, and Y. Zhao, "Network service analysis based on feature selection using improved linear mixed model," in *Communications, Signal Processing, and Systems: Proceedings of the 8th International Conference on Communications, Signal Processing, and Systems 8th*, Springer, 2020, pp. 2581–2589.

[101] A. Morichetta, P. Casas, and M. Mellia, "Explain-it: Towards explainable ai for unsupervised network traffic analysis," in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, 2019, pp. 22–28.

[102] M. M. Mampaka and M. Sumbwanyambe, "Poor data throughput root cause analysis in mobile networks using deep neural network," in *2019 IEEE 2nd Wireless Africa Conference (WAC)*, IEEE, 2019, pp. 1–6.

[103] A. Ghasemi, "Data-driven prediction of cellular networks coverage: An interpretable machine-learning model," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2018, pp. 604–608.

[104] C. Fiandrino, G. Attanasio, M. Fiore, and J. Widmer, "Toward native explainable and robust ai in 6g networks: Current state, challenges and road ahead," *Computer Communications*, vol. 193, pp. 47–52, 2022.

[105] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Deepcog: Optimizing resource provisioning in network slicing with ai-based capacity forecasting," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 361–376, 2019.

[106] K. Zhu and E. Hossain, "Virtualization of 5G cellular networks as a hierarchical combinatorial auction," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2640–2654, 2015.

[107] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, "Reinforcement learning for slicing in a 5g flexible ran," *Journal of Lightwave Technology*, vol. 37, no. 20, pp. 5161–5169, 2019.

[108] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1455–1470, 2019.

[109] C. Gutterman, E. Grinshpun, S. Sharma, and G. Zussman, "Ran resource usage prediction for a 5g slice broker," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 231–240.

[110] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Aztec: Anticipatory capacity allocation for zero-touch network slicing," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, IEEE, 2020, pp. 794–803.

[111] X. Shen *et al.*, "Ai-assisted network-slicing based next-generation wireless networks," *IEEE Open Journ. Vehic. Tech.*, vol. 1, pp. 45–66, 2020.

[112] F. Malandrino and C.-F. Chiasserini, "5G traffic forecasting: If verticals and mobile operators cooperate," in *2019 15th Ann. Conf. Wire. On-dem. Net. Sys. Serv. (WONS)*, IEEE, 2019, pp. 79–82.

[113] Q. Guo *et al.*, "Proactive dynamic network slicing with deep learning based short-term traffic prediction for 5g transport network," in *2019 Opt. Fibe. Commun. Conf. Exhib. (OFC)*, IEEE, 2019, pp. 1–3.

[114] G. Sun *et al.*, "Dynamic reservation and deep reinforcement learning based autonomous resource slicing for virtualized radio access networks," *IEEE Access*, vol. 7, pp. 45 758–45 772, 2019.

[115] V. Sciancalepore, F. Z. Yousaf, and X. Costa-Perez, "Z-torch: An automated nfv orchestration and monitoring solution," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1292–1306, 2018.

[116] Y. L. Lee, J. Loo, T. C. Chuah, and L.-C. Wang, "Dynamic network slicing for multi-tenant heterogeneous cloud radio access networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2146–2161, 2018.

[117] M. R. Raza, M. Fiorani, A. Rostami, P. Öhlen, L. Wosinska, and P. Monti, "Dynamic slicing approach for multi-tenant 5G transport networks," *Journal of Optical Communications and Networking*, vol. 10, no. 1, A77–A90, 2018.

[118] G. Tseliou, K. Samdanis, F. Adelantado, X. C. Pérez, and C. Verikoukis, "A capacity broker architecture and framework for multi-tenant support in LTE-A networks," in *2016 IEEE International Conference on Communications (ICC)*, IEEE, 2016, pp. 1–6.

[119] A. Baumgartner, T. Bauschert, A. A. Blzarour, and V. S. Reddy, "Network slice embedding under traffic uncertainties—a light robust approach," in *13th International Conference on Network and Service Management (CNSM)*, IEEE, 2017, pp. 1–5.

[120] G. Wang, G. Feng, W. Tan, S. Qin, R. Wen, and S. Sun, "Resource allocation for network slices in 5G with network resource pricing," in *GLOBECOM IEEE Global Communications Conference*, IEEE, 2017, pp. 1–6.

[121] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing in 5G: An auction-based model," in *IEEE International Conference on Communications (ICC)*, IEEE, 2017, pp. 1–6.

[122] K. Zhu and E. Hossain, "Virtualization of 5G cellular networks as a hierarchical combinatorial auction," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2640–2654, 2015.

[123] Y. Zhang, S. Bi, and Y.-J. A. Zhang, "Joint spectrum reservation and on-demand request for mobile virtual network operators," *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 2966–2977, 2018.

[124] A. Terra *et al.*, "Explainability methods for identifying root-cause of sla violation prediction in 5G network," in *IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2020, pp. 1–7.

[125] W. Guo, "Partially explainable big data driven deep reinforcement learning for green 5g uav," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, 2020, pp. 1–7.

[126] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[127] IBM, "Cost of a Data Breach Report 2022," Tech. Rep., 2022. [Online]. Available: `https://www.ibm.com/downloads/cas/3R8N1DZJ` (visited on 01/08/2023).

[128] Mandiant, *Zero Tolerance: More Zero-Days Exploited in 2021 Than Ever Before*, 2022. [Online]. Available: `https://www.mandiant.com/resources/blog/zero-days-exploited-2021` (visited on 01/08/2023).

[129] L. N. Tidjon, M. Frappier, and A. Mammar, "Intrusion detection systems: A cross-domain overview," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3639–3681, 2019.

[130] M. Panda and M. R. Patra, "Network intrusion detection using naive bayes," *International Journal of Computer Science and Network Security*, vol. 7, no. 12, pp. 258–263, 2007.

[131] J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, 2008.

[132] B. Ingre, A. Yadav, and A. K. Soni, "Decision tree based intrusion detection system for NSL-KDD dataset," in *Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 2 2*, Springer, 2018, pp. 207–218.

[133] A. Chandrasekhar and K Raghuveer, "Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers," in *2013 International Conference on Computer Communication and Informatics*, IEEE, 2013, pp. 1–7.

[134] M. Moradi and M. Zulkernine, "A neural network based system for intrusion detection and classification of attacks," in *Proceedings of the IEEE International Conference on Advances in Intelligent Systems*, IEEE Lux-embourg-Kirchberg, Luxembourg, 2004, pp. 15–18.

[135] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in *IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE, 2017, pp. 313–316.

[136] B. S. Bhati, G. Chugh, F. Al-Turjman, and N. S. Bhati, "An improved ensemble based intrusion detection technique using XGBoost," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, e4076, 2021.

[137] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *2016 Inter. Conf. Wire. Netw. Mobile Comm. (WINCOM)*, IEEE, 2016, pp. 258–263.

[138] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comp. Intel.*, vol. 2, no. 1, pp. 41–50, 2018.

[139] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[140] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42 169–42 184, 2020.

[141] J. R. Goodall, E. D. Ragan, C. A. Steed, *et al.*, "Situ: Identifying and explaining suspicious behavior in networks," *IEEE Trans. Visual. Comp. Graphics*, vol. 25, no. 1, pp. 204–214, 2018.

[142] S. Hariharan, A. Velicheti, A. Anagha, C. Thomas, and N Balakrishnan, "Explainable artificial intelligence in cybersecurity: A brief review," in *2021 4th International Conference on Security and Privacy (ISEA-ISAP)*, IEEE, 2021, pp. 1–12.

[143] F. Charmet, H. C. Tanuwidjaja, S. Ayoubi, *et al.*, "Explainable artificial intelligence for cybersecurity: A literature survey," *Annals of Telecommunications*, vol. 77, no. 11-12, pp. 789–812, 2022.

[144] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An explainable machine learning framework for intrusion detection systems," *IEEE Access*, vol. 8, pp. 73 127–73 141, 2020.

[145] H. Liu, C. Zhong, A. Alnusair, and S. R. Islam, "Faixid: A framework for enhancing ai explainability of intrusion detection results using data cleaning techniques," *Journal of Network and Systems Management*, vol. 29, no. 4, p. 40, 2021.

[146] K. Amarasinghe, K. Kenney, and M. Manic, "Toward explainable deep neural network based anomaly detection," in *2018 11th Int. Conf. Human Syst. Inter. (HSI)*, IEEE, 2018, pp. 311–317.

[147] 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*, 2018, pp. 364–379.

[148] M. Keshk, N. Koroniotis, N. Pham, N. Moustafa, B. Turnbull, and A. Y. Zomaya, "An explainable deep learning-enabled intrusion detection framework in iot networks," *Information Sciences*, vol. 639, p. 119 000, 2023.

[149] R. Younisse, A. Ahmad, and Q. Abu Al-Haija, "Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP)," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 126, 2022.

[150] K. Fujita, T. Shibahara, D. Chiba, M. Akiyama, and M. Uchida, "Objection!: Identifying Misclassified Malicious Activities with XAI," in *ICC 2022-IEEE International Conference on Communications*, IEEE, 2022, pp. 2065–2070.

[151] J.-B. Monteil, J. Hribar, P. Barnard, Y. Li, and L. A. DaSilva, "Resource reservation within sliced 5g networks: A cost-reduction strategy for service providers," in

2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020, pp. 1–6.

[152] S. M. Lundberg, G. Erion, H. Chen, *et al.*, "Explainable ai for trees: From local explanations to global understanding," *arXiv preprint arXiv:1905.04610*, 2019.

[153] J. Adebayo *et al.*, "Debugging tests for model explanations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 700–712.

[154] M. Abadi, A. Agarwal, *et al.*, *TensorFlow: Large-scale machine learning on heterogeneous systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[155] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, *et al.*, *Keras Tuner*, `https://github.com/keras-team/keras-tuner`, 2019.

[156] F. A. Silva, A. C. Domingues, and T. R. B. Silva, "Discovering mobile application usage patterns from a large-scale dataset," *ACM Trans. Knowl. Discov. Data (TKDD)*, vol. 12, no. 5, pp. 1–36, 2018.

[157] S. Li, E. Magli, G. Francini, and G. Ghinamo, "Deep learning based prediction of traffic peaks in mobile networks," *Computer Networks*, vol. 240, p. 110 167, 2024.

[158] A. Adams and P. Vamplew, "Encoding and decoding cyclic data," *The South Pacific Journal of Natural Science*, vol. 16, 1998.

[159] Y. Bassil, *A comparative study on the performance of permutation algorithms*, 2012. arXiv: `1205.2888`.

[160] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[161] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[162] F. B. Kokulu, A. Soneji, T. Bao, *et al.*, "Matched and mismatched socs: A qualitative study on security operations center issues," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 1955–1970.

[163] M. Vielberth, F. Böhm, I. Fichtinger, and G. Pernul, "Security operations center: A systematic study and open challenges," *Ieee Access*, vol. 8, pp. 227 756–227 779, 2020.

[164] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symp. Comp. Intel. Secur. Def. App.*, IEEE, 2009, pp. 1–6.

[165] CERT/NetSA, Carnegie Mellon University. "Silk (system for internet-level knowledge)." Accessed: February 2023. (), [Online]. Available: `http://tools.netsa.cert.org/silk`.

[166] S. Burschka and B. Dupasquier, "Tranalyzer: Versatile high performance network traffic analyser," in *IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2016, pp. 1–8.

[167] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Encrypted and VPN Traffic using Time-related Features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, 2016, pp. 407–414.

[168] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 4, pp. 1–21, 2012.

[169] C. Benzaid and T. Taleb, "Ai-driven zero touch network and service management in 5g and beyond: Challenges and research directions," *IEEE Network*, vol. 34, no. 2, pp. 186–194, 2020.

[170] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journ. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[171] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," *EAI Endor. Trans. Secur. Safety*, vol. 3, no. 9, e2, 2016.

[172] D. L. Marino, C. S. Wickramasinghe, and M. Manic, "An adversarial approach for explainable ai in intrusion detection systems," in *IECON 2018-44th Ann. Conf. IEEE Indust. Elect. Soc.*, IEEE, 2018, pp. 3237–3243.

[173] R.-F. Samoilescu, A. Van Looveren, and J. Klaise, "Model-agnostic and scalable counterfactual explanations via reinforcement learning," *arXiv preprint arXiv:2106.02597*, 2021.

[174] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, pp. 1–25, 2022.

[175] P. Barnard, N. Marchetti, and L. A. DaSilva, "Robust network intrusion detection through explainable artificial intelligence (xai)," *IEEE Networking Letters*, vol. 4, no. 3, pp. 167–171, 2022.

[176] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, vol. 1, pp. 108–116, 2018.

[177] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei, and F. Sabrina, "Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset," *IEEE Access*, vol. 9, pp. 140 136–140 146, 2021.

[178] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *IEEE Symposium on Security and Privacy*, IEEE, 2010, pp. 305–316.

[179] C. I. for Cybersecurity, *A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)*, 2018. [Online]. Available: `https://registry.opendata.aws/cse-cic-ids2018` (visited on 12/01/2022).

[180] A. Sirisha, K. Chaitanya, K. Krishna, and S. S. Kanumalli, "Intrusion detection models using supervised and unsupervised algorithms-a comparative estimation," *International Journal of Safety and Security Engineering*, vol. 11, no. 1, pp. 51–58, 2021.

[181] P. Dini, A. Begni, S. Ciavarella, *et al.*, "Design and testing novel one-class classifier based on polynomial interpolation with application to networking security," *IEEE Access*, vol. 10, pp. 67 910–67 924, 2022.

[182] R. Panigrahi and S. Borah, "A detailed analysis of cicids2017 dataset for designing intrusion detection systems," *International Journal of Engineering & Technology*, vol. 7, no. 3.24, pp. 479–482, 2018.

[183] V. Bulavas, V. Marcinkevičius, and J. Rumiński, "Study of multi-class classification algorithms' performance on highly imbalanced network intrusion datasets," *Informatica*, vol. 32, no. 3, pp. 441–475, 2021.

[184] K. P. Murphy, "Machine learning: A probabilistic perspective (adaptive computation and machine learning series)," *The MIT Press*, 2018.

[185] S. Milani, N. Topin, M. Veloso, and F. Fang, "A survey of explainable reinforcement learning," *arXiv preprint arXiv:2202.08434*, 2022.

[186] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[187] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106 685, 2021.