

S P A T I A L

D2.1 EXISTING AI ALGORITHMS AND THEIR ACCOUNTABILITY AND RESILIENCE FEATURES WITHIN THE CONTEXT OF APPLICATIONS TO IoT, 5G, AND CYBERSECURITY

Revision: v.1.0

Work package	WP2
Task	Task 2.1
Due date	30/06/2022
Submission date	30/06/2022
Deliverable lead	Fraunhofer Institute for Open Communication Systems (FOKUS)
Version	1.0
Authors <i>(sorted by alphabetical order of partner names)</i>	Michell Boerger (FOKUS), Helene Knof (FOKUS), Denis Rangelov (FOKUS), Lisa-Maria Schottstädt (FOKUS), Nikolay Tcholtchev (FOKUS), Samuel Marchal (FSC), Dusan Borovcanin (MFX), Ana Cavalli (MI), Vinh Hoa La (MI), Manh Dung Nguyen (MI), Prachi Bagave (TUD), Aaron Ding (TUD), Madhusanka Liyanage (UCD), Chamara Sandeepa (UCD), Thulitha Senevirathna (UCD), Bartłomiej Siniarski (UCD), Shen Wang (UCD), Abdul-Rasheed Ottun (UT)
Reviewers	Huber Flores (UT) Souneil Park (TID)



Grant Agreement No.: 101021808
 Call: H2020-SU-DS-2020
 Topic: SU-DS02-2020
 Type of Action: RIA

D2.1: Accountability and Resilience Analysis

<p>Abstract</p>	<p>To establish a foundation for the research activities towards a SPATIAL platform, the present document seeks to understand the accountability and resilience of existing Machine Learning (ML) algorithms. Thereby, the selection of ML algorithms analysed in this document is based on their potential application in the four SPATIAL use cases, which reflect the domains Internet of Things (IoT), 5G, cybersecurity, and eHealth. In order to acquire the above-mentioned understanding, this deliverable analyses the identified relevant ML algorithms concerning their accountability, explainability and resilience characteristics. Thereby, different Explainable AI (XAI) methods are taken into account and their applicability to the ML algorithms in question is discussed. Furthermore, the selected ML algorithms are looked into with regard to their resilience against modern attacks on the AI models (e.g., data poisoning, model stealing, evasion attacks etc.).</p>
<p>Keywords</p>	<p>Explainable AI, Resilient AI, Robust AI, Trustworthy AI, Secure AI, Explainability, Accountability, Resilience, Adversarial Attacks, Analysis, Machine Learning</p>

Document Revision History

Version	Date	Description of change	List of contributor(s)
v0.1	02/11/2021	Initial ToC	Nikolay Tcholtchev Michell Boerger
v0.2	21/05/2022	Integrated contributions from all involved partners.	All listed authors
v0.3	31/05/2022	Prepared document for internal review.	Nikolay Tcholtchev Michell Boerger
v0.4	21/06/2022	Addressed comments from internal reviewers	Shen Wang Michell Boerger
v1.0	28/06/2022	Prepared document for final submission	Michell Boerger Nikolay Tcholtchev

DISCLAIMER

The information, documentation and figures available in this deliverable are written by the SPATIAL project's consortium under EC grant agreement 101021808 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

COPYRIGHT NOTICE

© 2021 - 2024 SPATIAL



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

Project funded by the European Commission in the H2020 Programme		
Nature of the deliverable:		R*
Dissemination Level		
PU	Public, fully open, e.g., web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to SPATIAL project and Commission Services	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

EXECUTIVE SUMMARY

This deliverable document is part of the initial steps of the SPATIAL project. Within this starting phase, the goal is to capture the requirements and general design principles for modern system architectures based on accountable Artificial Intelligence (AI). A further goal is to propose resilient accountability metrics and embed them into existing AI algorithms. To establish a foundation for this ambition, the present document seeks to understand the accountability and resilience of existing Machine Learning (ML) algorithms. Thereby, the selection of ML algorithms analysed is based on the SPATIAL use cases, which reflect the domains IoT, 5G, cybersecurity, and eHealth. In order to acquire the above-mentioned understanding, the identified ML algorithms were analysed concerning their accountability and resilience characteristics.

Since the SPATIAL project understands explainability as a means to achieve accountability, the accountability analysis focuses on the explainability of the ML algorithms. Precisely, the intrinsic explainability of their underlying algorithmic properties was examined. However, since the findings showed that many of the discussed models are non-comprehensible and non-transparent black-boxes, the applicability of various Explainable AI (XAI) methods was also analysed. The analysis revealed that the identified state-of-the-art XAI methods can indeed be used to improve the local and global explainability of black-box models, thus enhancing their accountability. However, it must be mentioned that determining the most appropriate XAI method for an algorithm cannot be done a-priori, since it depends on both the task at hand and the user to whom the explanations are addressed.

Regarding the resilience analysis, the ML algorithms were examined in terms of their vulnerability to adversarial ML attacks (e.g., poisoning attacks, evasion attacks, data inference attacks, and model stealing attacks). More specifically, recent scientific literature that has studied the algorithms' vulnerability to these attacks was identified. The obtained findings indicate that all ML algorithms discussed are to some degree vulnerable to the studied adversarial attacks. This broadens the attack surface and introduces new vulnerabilities and security risks for ML-based systems. For example, it was identified that adversarial attacks could cause significant degradation of model performance (poisoning attacks), serious operational issues (evasion attacks), privacy issues (data inference attacks), and violations of intellectual properties (model stealing attacks). In addition, there are indications that the attacker's success rate depends on their knowledge of the specifics of the ML model - in this context white-box models are more prone to adversarial attacks than black-box models. The analysis also indicated that the attacker's success rate is subject to the used dataset and the concrete application domain. This suggests that application-independent comparability of the vulnerability of ML models is difficult, which implies that no general statements can be made about the degree of vulnerability of the models.

In conclusion, it can be stated that discussed ML algorithms hold different resilience and accountability characteristics. Furthermore, the findings suggest that selecting a suitable ML algorithm always constitutes a trade-off between performance, accountability, and resilience. Typically, the higher the performance of an ML algorithm, the less accountability it offers. The problem of finding an optimal balance for this trade-off clearly demonstrates the need for appropriate measures to compare and assess the accountability, resilience, and accuracy of ML models.



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	4
TABLE OF CONTENTS	5
LIST OF FIGURES	7
LIST OF TABLES	8
ABBREVIATIONS	9
1 INTRODUCTION	11
1.1 Scope and Objectives of the Deliverable.....	12
1.2 Structure of the Deliveralbe.....	13
2 STATE OF THE ART	14
2.1 Review of relevant Terms on Deliverable objectives.....	14
2.1.1 Explainability.....	14
2.1.2 Accountability	15
2.1.3 Resilience	15
2.2 Artificial Intelligence Algorithms for SPATIAL use cases.....	16
2.2.1 (Deep) Neural Networks	16
2.2.2 Support Vector Machines.....	18
2.2.3 Decision Trees	20
2.2.4 Random Forests	21
2.2.5 Gradient Boosted Trees and XGBOOST	22
2.2.6 Bayesian Networks.....	24
2.3 Explainable AI Methods.....	25
2.3.1 LIME.....	25
2.3.2 SHAP	26
2.3.3 Counterfactual Explanations.....	27
2.3.4 Permutation Feature Importance	28
2.3.5 Partial Dependence Plot.....	29
2.3.6 t-SNE	30
2.3.7 Layer-wise Relevance Propagation	32
2.3.8 Occlusion Sensitivity.....	34
2.3.9 CAM and Grad CAM.....	36



SPATIAL project is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 101021808.

3	ACCOUNTABILITY AND RESILIENCE ANALYSIS OF IDENTIFIED AI ALGORITHMS	39
3.1	Analysis of identified AI Algorithms.....	39
3.1.1	(Deep) Neural Networks	40
3.1.2	Support Vector Machines.....	42
3.1.3	Decision Trees	43
3.1.4	Random Forests	45
3.1.5	Gradient-boosted trees and XGBoost.....	47
3.1.6	Bayesian Networks.....	48
3.2	Accountability and Resilience Requirements for the SPATIAL Use Cases	50
3.2.1	Use Case 1: Privacy-preserving AI on the Edge and beyond.....	50
3.2.2	Use Case 2: Improving explainability, resilience and performance of cybersecurity analysis of 4G/5G/IoT Networks.....	51
3.2.3	Use Case 3: Accountable AI in Emergency eCall System.....	52
3.2.4	Use Case 4: Resilient Cybersecurity Analytics.....	54
4	DISCUSSION AND RECOMMENDATIONS.....	56
4.1	Findings on Accountability	56
4.2	Findings on Resilience	57
5	CONCLUSIONS AND OUTLOOK.....	59
	REFERENCES	61



LIST OF FIGURES

FIGURE 1: ARTIFICIAL NEURON ARCHITECTURE 16

FIGURE 2: MULTILAYER ANN ARCHITECTURE (CREATED WITH HTTP://ALEXLENAIL.ME/NN-SVG/INDEX.HTML)..... 17

FIGURE 3: SVM EXAMPLE 19

FIGURE 4: EXAMPLE FOR THE USAGE OF THE "KERNEL TRICK" 20

FIGURE 5: DECISION TREE ARCHITECTURE 21

FIGURE 6: RANDOM FOREST ARCHITECTURE 22

FIGURE 7: A DAG GRAPH REPRESENTING TWO INDEPENDENT CAUSES OF COMPUTER FAILURE [23] 24

FIGURE 8: SHUFFLING THE VALUES OF HEIGHT AT AGE 10 FOR PERMUTATION IMPORTANCE CALCULATION [35] 28

FIGURE 9: PARTIAL DEPENDENCE 2D/3D PLOTS FOR THE CALIFORNIA HOUSING DATASET [38] 30

FIGURE 10: T-SNE APPLIED ON THE MNIST DATASET [39]..... 31

FIGURE 11: LRP EXAMPLE (GENERATED WITH HTTPS://LRPSERVER.HHI.FRAUNHOFER.DE/IMAGE-CLASSIFICATION)..... 32

FIGURE 12: BACKPROPAGATION OF RELEVANCE FROM THE OUTPUT TO THE INPUT LAYER (TAKEN FROM [43]) 33

FIGURE 13: LRP METHOD VISUALISED (TAKEN WITHOUT CHANGES FROM [45]) 34

FIGURE 14: GRAPHICAL PRODUCE OF OCCLUSION SENSITIVITY METHOD (AUTHORS' OWN CONTRIBUTION)..... 35

FIGURE 15: OCCLUSION SENSITIVITY FOR INTERPRETABILITY FOR MEDICAL IMAGING IN IOT APPLICATIONS (AUTHORS' OWN CONTRIBUTION USING THE TOOL [118]) 36

FIGURE 16: CLASS ACTIVATION MAPPING: CAM HIGHLIGHTING THE DISCRIMINATIVE IMAGE REGIONS USED BY CNN FOR CLASSIFYING THE AUSTRALIAN TERRIER [49] 37



LIST OF TABLES

TABLE 1: OVERVIEW OF THE APPLICABLE XAI METHODS OF THE IDENTIFIED ML ALGORITHMS 40

TABLE 2: IDENTIFIED REFERENCES OF THE RESPECTIVE ADVERSARIAL ATTACKS AGAINST THE ML ALGORITHMS..... 42



SPATIAL project is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 101021808.

ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Network
CAM	Class Activation Mapping
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DWMRI	Diffusion Weighted Magnetic Resonance Imaging
EC	European Commission
eCall	Emergency Call
EMYNOS	nExt generation eMergencY commuNicatiOnS
EU	European Union
FL	Federated Learning
FOKUS	Fraunhofer Institute for Open Communication Systems
FSC	F-Secure OYJ
GAP	Global Average Pooling
GBT	Gradient-boosted tree
Grad-CAM	Gradient-weighted Class Activation Mapping
ICT	Information and Communications Technology
IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
k-SVM	kernel trick SVM
KPI	Key Performance Indicator
LIME	Local Interpretable Model-agnostic Explanations
LRP	Layer-Wise Relevance Propagation
MFx	Mainflux Labs
MI	Montimage EURL
MIA	Membership Interference Attack
ML	Machine Learning
MMT	Montimage Monitoring Tool



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

NG112	Next Generation 112
NIDS	Network Intrusion Detection System
NN	Neural Network
PDP	Partial Dependence Plot
RAN	Radio Access Network
RCA	Root Cause Analysis
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
SDN	Software Defined Network
SGD	Stochastic Gradient Decent
SNE	Stochastic Neighbor Embedding
SHAP	Shapley Additive Explanations
SVM	Support Vector Machine
t-SNE	t-distributed Stochastic Neighbor Embeddings
TID	Telefonica Investigacion Y Desarrollo SA
TLS	Transport Layer Security
TUD	Delf University of Technology
UCD	University College Dublin
UT	University of Tartu
VoIP	Voice over IP
XAI	Explainable AI
XGBoost	eXtreme Gradient Boosting



1 INTRODUCTION

For digital applications in critical infrastructure domains such as energy, emergency communication, cyber security, Internet of Things (IoT), 5G, automotive, or railway, the security and accountability of the deployed applications and systems are of critical importance. Specifically, in the European context, certification and verification of the underlying functionality of the developed algorithms and applications are often mandatory prior to their deployment in critical infrastructures. In this context, it must be guaranteed that such systems can be made accountable for their behaviour and are secure, safe, and sufficiently resilient to cyberattacks. These accountability and security requirements imply the need for a transparent understanding of the underlying functionality of such systems and the necessity to characterize their resilience. Several established formal methods already exist for the analysis, verification, and certification of traditional systems (e.g., rule-based expert systems), which are accepted in both industry and academia and enable conclusions about the accountability and resilience of such systems. In contrast, in the domain of Artificial Intelligence (AI), especially in the field of Machine Learning (ML), it is still difficult to assess or even achieve accountability and resilience of ML-based systems. However, this reveals a conflict with the latest technological developments. In recent years, the ML field has gained much attention in the industrial and academic context. Many practical ML applications could demonstrate the great potential and the superior performance of ML-based applications and systems in terms of efficiency and accuracy compared to their traditional equivalents. As a result, ML-based applications and systems are finding applications in more diverse domains, including the critical infrastructure domains mentioned above.

However, the integration of ML algorithms raises several new challenges regarding the accountability of ML-based systems. In the context of AI, we understand accountability as the representation of the AI models in a way that they can be easily understood. Nevertheless, many widely used high-performing ML algorithms like Deep Neural Networks (DNNs) or random forests (RF) suffer from a lack of transparency and explainability. Due to their sheer complexity, the decision-making process and the underlying functioning of many ML algorithms and techniques can no longer be comprehended and understood by human operators or auditors. As a result, such ML algorithms are perceived as opaque black-boxes¹ in which the decision making and the underlying reasoning behind it remains non-transparent. This also affects the understanding of the behaviour of the ML-based systems, which leads to questions about the accountability of the ML algorithms and calls for methods to understand such algorithms and their decisions.

Therefore, the research area of so-called Explainable AI (XAI) has been established in recent years that aims to explore solutions to this problem. This research area tries to develop methods to explain individual decisions of ML algorithms as well as to provide a global understanding of the functioning of entire models. The former is typically referred to as local explainability,

¹ The terms "black box" and "white box" can be understood as offensive and exclusionary terminology. Therefore, the use of non-discriminatory synonyms such as "opaque box" and "clear box" is recommended [119]. However, since the original terms are still heavily used in the technical community, we will also continue to use these in this document. Nevertheless, we would like to clearly distance ourselves from any form of discrimination and racism.



D2.1: Accountability and Resilience Analysis

whereas the latter is denoted as global explainability. In the context of the SPATIAL project, XAI is particularly relevant since we understand the explainability of ML algorithms as means to achieve accountability. In this regard, it is essential to understand and analyse which XAI methods exist, which inherent accountable characteristics black-box ML models possess, and how the accountability can be improved by the explainability provided through XAI methods. Such an analysis represents a first objective of this deliverable.

Besides the limited accountability discussed above, the integration of ML algorithms into traditional systems (e.g., rule-based expert systems) also raises new security concerns and challenges with respect to the resilience of ML-based systems. More precisely, the ML algorithms can become targets of adversarial ML attacks, which broadens the attack surface and introduce new vulnerabilities and security risks for ML-based systems. This calls for a clear understanding and analysis of the resilience of ML algorithms to adversarial attacks, in order to be able to develop and apply appropriate countermeasures to secure ML-based systems. Such analysis constitutes another objective of this document.

1.1 SCOPE AND OBJECTIVES OF THE DELIVERABLE

The SPATIAL project plans to tackle the above-mentioned gaps and challenges of black-box AI by designing and developing resilient accountable metrics, privacy-preserving methods, verification tools, and system solutions that will serve as critical building blocks for trustworthy AI in Information and Communications Technology (ICT) systems and cybersecurity. In this context, the project covers data privacy, resilience engineering, and legal-ethical accountability toward trustworthy AI. This is intended to support the European Union (EU) in its ambition to be at the forefront of accountability and resilience AI, and accelerate its efforts in the evolution towards trustworthy AI.

This deliverable document is assigned to the initial phase of the SPATIAL project. This initial phase aims to capture the requirements and general design principles for modern system architectures based on accountable AI. A further goal is to propose resilient accountability metrics and embed them into the existing AI algorithms. To establish a foundation for this ambition, the present document seeks to understand the accountability and resilience of existing ML algorithms. Hence, we will discuss existing relevant ML algorithms with respect to their accountability and resilience characteristics. Since we see explainability as a means to achieve accountability in SPATIAL, we will also examine the algorithms concerning the applicability of XAI methods.

Regarding the resilience analysis of the ML algorithms, we will examine them in terms of their vulnerability to adversarial ML attacks. The analysis conducted in this document will form the basis to improve the explainability and resilience of ML algorithms and the development of the envisioned accountability metrics and their integration into the existing AI algorithms. To summarize, this deliverable document will provide the following contributions:

- review of the terms explainability, accountability, and resilience, in order to establish a common understanding in the context of the deliverable objectives
- brief introduction to the theoretical foundations of six widely used ML algorithms that will also be potentially used in the four SPATIAL use cases, namely DNNs, Support



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

- Vector Machines (SVMs), decision trees, random forests, gradient-boosted trees (GBTs) and eXtreme Gradient Boosting (XGBoost), and Bayesian networks
- comprehensive overview and discussion of state-of-the-art XAI methods that provide both local and global explainability
 - analysis of the accountability and explainability of the six ML algorithms mentioned above and an identification of which XAI methods are applicable to which ML algorithm (see Table 1)
 - resilience analysis of the discussed ML algorithms regarding their vulnerability to adversarial ML attacks (see Table 2)

1.2 STRUCTURE OF THE DELIVERABLE

The remainder of the deliverable document at hand is comprised of four additional sections. This introduction section is followed by Section 2, in which we will present the necessary theoretical background for this document. More precisely, we will review relevant terms and provide theoretical foundations for the six ML algorithms analysed in this deliverable. In addition, we will examine some relevant XAI methods that can be used to provide local and global explanations for the discussed black-box ML models. Subsequently, Section 3 constitutes the main section of this document. We begin Section 3 by summarizing the four SPATIAL use cases and highlighting their need for accountable and resilient ML algorithms. Afterwards, we will analyse the identified ML algorithms with respect to their accountability and resilience characteristics. Based on this, we will briefly discuss the findings and derive recommendations in Section 4. Finally, we will conclude the deliverable document in Section 5.



2 STATE OF THE ART

In the following, we will review the theoretical background essential for interpreting the contents of the present document. First, we will review relevant terms about this deliverable objectives to build a common understanding. Then, we will present the six ML algorithms that will be analysed. The sub-selection describing the algorithms is based on their potential application in the four SPATIAL use cases. Finally, we present relevant XAI methods that can be applied to the identified ML algorithms to support their explainability and thus achieve accountability.

2.1 REVIEW OF RELEVANT TERMS ON DELIVERABLE OBJECTIVES

The relevant terms discussed in this section summarize the definitions provided in the SPATIAL deliverable D1.1. We refer the interested reader to that deliverable for a detailed elaboration of the terms.

2.1.1 EXPLAINABILITY

Explainability in AI is regarded as the ability for a human to understand the decision-making process of a given model with the help of its feature space, training records, targets, and the ML algorithm itself [87]. Consequently, the model explanation should be credible and reliable enough to give trust to the user about the model's behaviour.

In the recent developments in the AI/ML field, many of the algorithms are complicated models that are designed to tackle complex tasks by identifying subtle patterns in large datasets. Although these algorithms perform extremely well in terms of accuracy, human-centric comprehensibility of the decision-making process inside those models is not always straightforward. A good example is a neural network. Due to its non-linear and complex modelling capabilities, the decision-making process of a neural network is not directly understandable to humans without the help of additional information [1]. Hence, a neural network and its decision-making process is perceived as a "black-box" for human operators. On the other hand, decision trees are generally accepted as inherently "transparent" models where the decisions are self-explanatory based on the algorithm and the dataset.

For the process of creating a model explanation, it is not enough to utilize transparent models that allow for an understanding of their inner workings and decision-making (e.g., decision trees). It also necessary to generate a model interpretation [2]. Generating interpretations often requires separate tools in addition to the ML algorithm. Therefore, the explainability of a system is based on the ability to generate an explanation, so that the human users can understand the relation between predictions and input data.

It's also worth pointing out that explainability of a system is relative to the target audience. A person with more technical knowledge would understand the decision process of an algorithm more clearly than a general high-level user.



2.1.2 ACCOUNTABILITY

Accountability is most widely accepted as “the obligation to explain and justify conduct” with an implicit warning that “*accountability is elusive*” [5]. It is often necessary when the entity in power does not behave as expected, causing a need to understand the reason behind the actions and identify the responsible person or organisation. Thus, ensuring accountability also inherently motivates actors to behave in a better way [7].

With growing AI applications, our society is going through radical changes. Our lives are getting interdependent on AI as it is growing in various sectors from medical sciences to household appliances. Recent AI failure incidents - like the fatality caused by Uber autonomous car [3], or the publicly made racist comments by Microsoft chat-bot Tay [6] - have elevated the concern of AI accountability. Thus, this has created a need to reason out the actions made by an AI, thereby creating a lot of attention in the research community to understand the black-box nature of AI.

Recently, such events have caused new developments and growing research interest in the explainable AI domain. In this area of research, AI developers and data scientists are trying to make the AI models more interpretable by explaining the decision-making process of the models. Additionally, the European Commission [4] has also enlisted accountability as one of the key requirements for AI development. The four major elements to ensure accountability are:

- **Auditability:** The systems should facilitate the ability to trace their actions
- **Minimizing and reporting negative impact:** The systems should ensure minimising the risks and reporting in the event of any mishaps.
- **Documenting trade-offs:** Any trade-offs to achieve accountability should be well documented.
- **Ability to redress:** In case of any accident, immediate corrective actions should be conducted.

2.1.3 RESILIENCE

AI systems are quickly becoming integrated into different critical components of cybersecurity systems, IoT and 5G networks. Organisations should ensure the resilience of their AI systems, similar to other mission-critical assets. AI systems will therefore be expected to operate in adversarial environments. Their ability to adapt to potential threats and risks, or their resilience, is indispensable and critical. The concept of “*resilient AI*” [8] encompasses the idea that AI systems continue to offer the intended services even in the presence of adversarial attacks, in order to guarantee safety and security of the systems in which they are deployed. The ability to resist adversarial attacks that compromise the integrity of AI systems, like poisoning attack during training and evasion attack during inference, is paramount for resilience. Resisting other adversarial attacks that compromise the confidentiality of AI systems, like model stealing attacks and data inference attacks, is also important but secondary in the context of resilience.

To ensure resilience of AI systems, we first need to be able to measure it. However, as current AI systems are still considered as black boxes due to a lack of explainability and transparency, assessing their resilience is different from common measures on non-AI systems. Indeed,



existing measurement approaches presume that humans are responsible for making any design decisions that may affect resilience. However, AI systems are trained using a huge amount of data collected from different resources that can exceed the capability of human operators to measure. Furthermore, the resilience of AI systems could be compromised by unintended issues in development and operational processes, malicious interactions with AI systems and the vulnerability (e.g., corruption, bias ...) of the training data, which have significant effects on the whole system’s performance. Some well-known examples of compromising the resilience of AI systems are serious accidents of self-driving vehicles [9] or unsafe and incorrect medical recommendations by IBM’s Watson [10].

2.2 ARTIFICIAL INTELLIGENCE ALGORITHMS FOR SPATIAL USE CASES

We will now present the theoretical foundation of the six ML algorithms analysed in this deliverable document. Again, we would like to emphasize that the selection of the algorithms presented here is grounded on their applicability in the four SPATIAL use cases.

2.2.1 (DEEP) NEURAL NETWORKS

One of the most common and widely used type of Machine Learning algorithms are the so-called Artificial Neural Networks (ANN). The reason behind their popularity is that they provide a very flexible, versatile, and scalable architecture, which in turn can be utilised for solving wide variety of problems with different levels of complexity. The most important building block of an ANN is the artificial neuron and its inner workings illustrated in Figure 1.

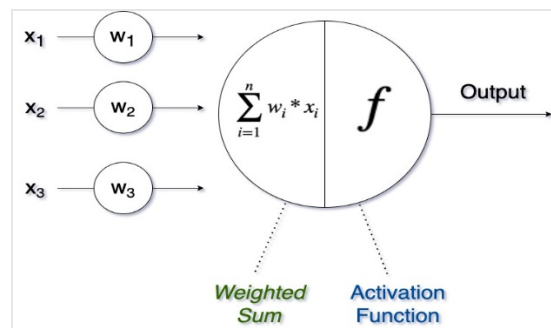


FIGURE 1: ARTIFICIAL NEURON ARCHITECTURE

Figure 1 shows a neuron that receives one or more inputs and for each of those, it assigns a corresponding weight value. The main function of the neuron is to (1) compute the weighted sum of the inputs and (2) to apply the so-called activation function on that sum. The output of this two-step process is the result produced by the neuron. In that setup the inputs correspond to specific features of the input data, while the weights express how “strongly” a particular input (i.e., feature) impacts the final output. Subsequently, the activation function determines whether the neuron should be “activated” or not. There are many different activation functions applied based on the task for which the ANN is used, but two of the most common ones include ReLU and sigmoid:



D2.1: Accountability and Resilience Analysis

$$\text{Relu}(x) = \max(0, x)$$

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$$

With this in mind, an architecture with only a single neuron is usually not powerful enough to learn and make predictions on complex tasks. To address this challenge, we can stack multiple neurons in multiple layers. Figure 2 illustrates this idea visually: As depicted in the diagram, in a Neural Network we have an input, output and hidden layers. The number of hidden layers depends on the complexity of the learned task. More complex tasks typically require more complex network architecture with additional hidden layers. Artificial Neural Networks with multiple (i.e., usually more than one) hidden layer are called “Deep Neural Networks” and they lie at the centre of the Deep Learning (DL) domain.

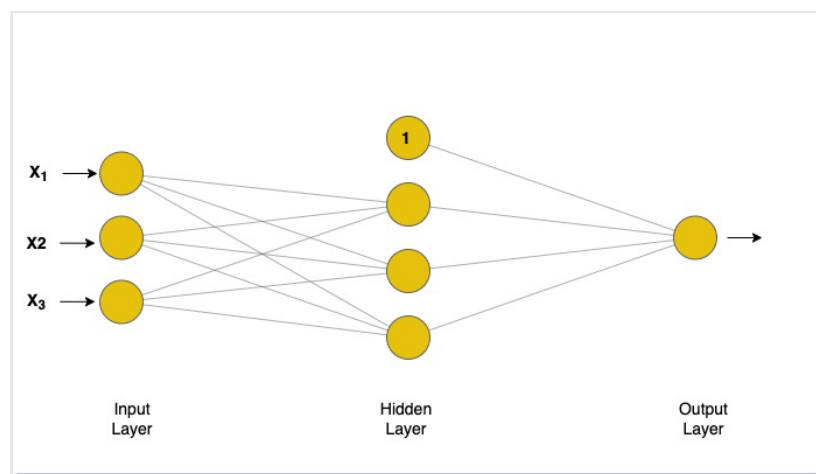


FIGURE 2: MULTILAYER ANN ARCHITECTURE (CREATED WITH [HTTP://ALEXLENAIL.ME/NN-SVG/INDEX.HTML](http://alexlenail.me/nn-svg/index.html))

Given the architecture of a Deep Neural Network, one important question remains unanswered and namely – “*how do these networks learn?*” In the context of Machine Learning, a model has successfully “learned” a task when the generated prediction error of the model on that task is reasonably small. The prediction error is computed by a “cost function” which measures how different is the predicted from the actual value. Put simply, the goal of an ML/DL practitioner is to minimize this cost function. This is typically achieved in a two-step process - the so-called forward and backward pass through the network. The forward pass (i.e., forward propagation) computes the neuron activations in the network in the direction from input to output layers and as a result the model generates a prediction. The backward pass (i.e., backward propagation) computes the cost function based on the generated prediction and then it computes the gradient of this cost function with respect to all network parameters (i.e. the network weights and biases). The main algorithm used for computing this gradient was proposed in 1986 by Rumelhart et al. [11] and is called “back propagation”. Back propagation allows us to efficiently compute the gradient of the cost function with respect to a specific network parameter. By doing so we can adjust this parameter (i.e., weight or bias) in a way that will minimize the network's cost function. This whole process is encompassed in an algorithm called “gradient descent”.



D2.1: Accountability and Resilience Analysis

In terms of versatility, DNNs can be used for a wide variety of use cases including both classification and regression tasks. However, important to note is that despite their versatility, DNN architectures in their traditional form have some limitations against highly specific tasks. For instance, two distinct domains that experienced significant innovation and progress are computer vision and natural language processing. These domains and their respective tasks have highly specific demands, which often times cannot be fulfilled by standard DNNs. This led to the introduction of new, more sophisticated Neural Network (NN) architectures such as Convolutional Neural Nets (CNNs) and Recurrent Neural Networks (RNNs).

CNNs utilize the so-called convolutional layers, which allow neurons to connect only to pixels from an image that belong to a specific region instead of connecting to all pixels in the image. This is especially helpful for image data, where we can have a large number of pixels and consequently a really slow training time.

RNNs introduce an architecture which allows us to work efficiently with sequential data such as time series, speech or text translation, where the current network output might depend on the prior elements of the sequence. Because of this dependence, the neurons in an RNN receive not only the input for the current time step but also the output from the previous time step. This architecture is more complex and requires a set of adjustments to the back propagation algorithm, in order to function as intended. Additionally, with the described architecture earlier inputs (i.e. inputs from earlier time steps) will gradually fade with each new time step. Therefore, a more sophisticated architecture such as long short-term memory was introduced [12].

2.2.2 SUPPORT VECTOR MACHINES

A Support Vector Machine is a supervised Machine Learning algorithm originally proposed by Boser, Guyon, and Vapnik in 1992 [13] SVMs can be used for linear and non-linear classification and regression tasks. The basic idea behind the SVM algorithm is to find a decision boundary that separates the data samples according to their label. The main idea behind the method used by SVMs for constructing the decision boundary is visualized in Figure 3. What the diagram shows is a decision boundary that can be described as an optimal hyperplane, since it separates the two classes with a maximum margin. More specifically, there is a multitude of hyperplanes that can separate the samples in Figure 3. However, SVMs try to find a decision boundary that maximizes the margin between the data samples from one class and data samples from the other classes. In this way, the trained SVM model should be able to generalize better to previously unseen instances.



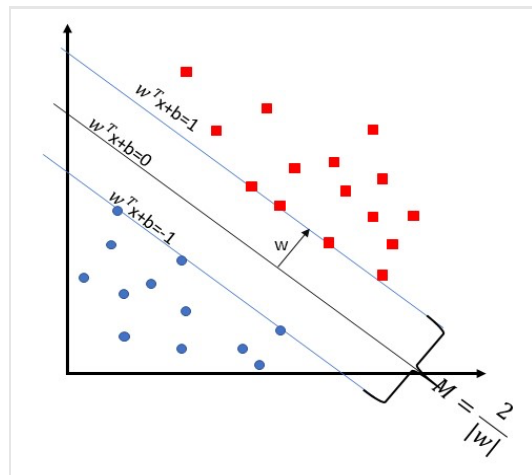


FIGURE 3: SVM EXAMPLE²

In order to find such maximum margin decision boundary, the SVM constructs a linear system which can be solved in either the primal or the dual form. This linear system is constructed in a way such that the data points must be on the correct side of the hyperplane with respect to their corresponding label. This is fulfilled when for all data points x_i with label $y_i \in \{-1, 1\}$ the following system of equations holds true [15]:

$$\begin{cases} w^T x_i + b \geq +1, & \text{if } y_i = +1 \\ w^T x_i + b \leq -1, & \text{if } y_i = -1 \end{cases}$$

These equations can also be presented in a more compact form as $y_i(w^T x_i + b) \geq 1$. Additionally, as illustrated in Figure 3, there are data points for which the equalities $w^T x_i + b = 1$ and $w^T x_i + b = -1$ hold true. Such data points are called support vectors and they lie directly on the hyperplanes. These points guide the position and the orientation of the hyperplane. In fact, based on these points, the SVM tries to satisfy a second set of constraints in an attempt to find the optimal decision boundary. This second set of constraints focuses on finding parameters w and b that satisfy the equation $y_i(w^T x_i + b) \geq 1$ (i.e., classifying all data samples correctly) while simultaneously maximizes the margin M which is defined as $M = \frac{2}{|w|}$. Discussing the exact details around the numerical solution of this optimization problem is out of the scope of this work.

With this in mind, while the description above focuses on how SVMs can be applied for linearly separable datasets, the algorithm is also useful for non-linearly separable use tasks. Nevertheless, in order to utilize an SVM for such non-linearly separable tasks, we have to apply the so-called **kernel trick** [15]. The idea behind the kernel trick is to use a kernel function that projects the data in a higher dimension, so that the SVM can be applied on this projection (see Figure 4). There are different types of popular kernel functions. Two of the most widely used ones include the Polynomial kernel and the Gaussian kernel. Choosing the most suitable kernel function depends on the input data, its properties and structure [14].

² Based and adapted from <https://dataaspirant.com/3-support-vector-machine-algorithm>



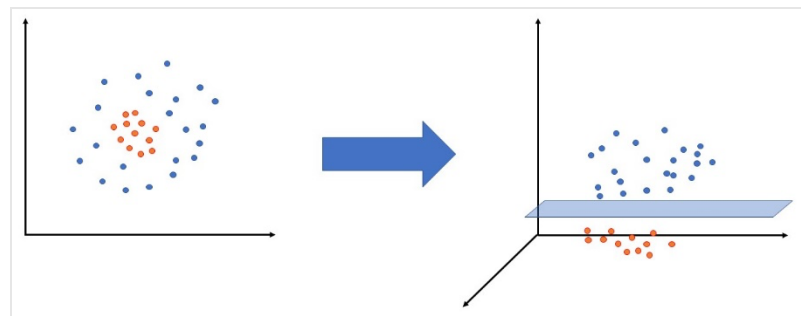


FIGURE 4: EXAMPLE FOR THE USAGE OF THE "KERNEL TRICK"³

Some of the primary reasons for using an SVM in comparison to other Machine Learning methods is that the algorithm is very versatile and can handle linear, non-linear, regression, and classification tasks. Additionally, SVMs generalize well to unknown data instances and tend to overfit less compared to other popular methods (e.g., decision trees).

2.2.3 DECISION TREES

A Decision Tree is a supervised Machine Learning algorithm that can be utilized for both regression and classification tasks. Figure 5 illustrates an example architecture for a decision tree applied on a classification task. As the diagram shows, the typical structure of a decision tree consists of a root, non-leaf and leaf nodes. The classification process starts at the root node which represents a standard conditional in the form - "if...then...else". The result of evaluating this conditional determines, which is the next node in the tree to be processed. After processing all child nodes in this manner, the end of the path is a leaf node which assigns the current data point the correct class. In other words, for a data sample to be classified, it has to go through the whole path from the root to one of the leaves [16].

The process of constructing an optimal decision tree is very computationally expensive. Therefore, in practice, many of the algorithms used for this process utilize greedy approaches. One such commonly used greedy method is Hunt's algorithm. The idea of this method is to recursively divide the data points into subsets until each subset consists of only datapoints with the same label. As long as there are two data points with the same label in the same subset, the algorithm finds an attribute, which splits the current subset into smaller subsets. The decision on how to split the subsets is performed based on a previously defined split condition. Some examples for such split conditions are the "Gini Index" and the "Information Gain" [17].

³ Based and adapted from https://www.researchgate.net/figure/Kernel-trick-By-transforming-the-original-space-left-into-a-space-of-increased-dimensions-fig1_305284381



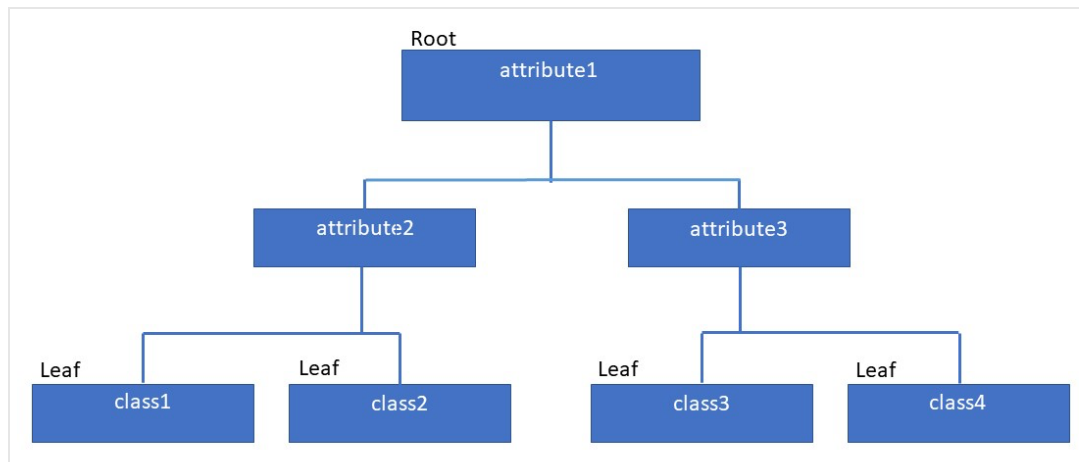


FIGURE 5: DECISION TREE ARCHITECTURE⁴

Some of the main advantages of decision trees is that they are relatively easy to interpret and are able to achieve high accuracy. However, decision trees also tend to overfit. This is especially true for large data sets where a very complex (i.e., large number of nodes and high depth) decision tree can memorize the patterns of the training data but would perform poorly for previously unseen instances. Additionally, standard decision trees are trained on a complete training set and are unable to incrementally adjust to new data instances. Instead, in the presence of new data samples, the decision tree has to be trained from scratch.

2.2.4 RANDOM FORESTS

A Random Forest is a supervised Machine Learning method proposed by L. Breiman in 2001 [19]. It is an ensemble algorithm that can be used for both classification and regression tasks and utilizes a group of Decision Trees, each of which is trained on a subset of the training data. The final prediction is generated as the aggregate of the predictions of the majority of the decision trees (see Figure 6). More specifically, when a new data sample has to be classified, each decision tree in the ensemble generates an independent prediction for that particular data point. The results generated by all trees in the ensemble are aggregated and used for establishing a majority voting that determines the final class prediction of the Random Forest. The same process can be applied for regression tasks, where the final predictions is computed as the average value of all decision trees [18] [19]. The main intuition behind this approach is to leverage the “knowledge” of multiple models instead of relying on a single source of truth.

⁴ Taken from <https://paragmali.me/building-a-decision-tree-classifier>



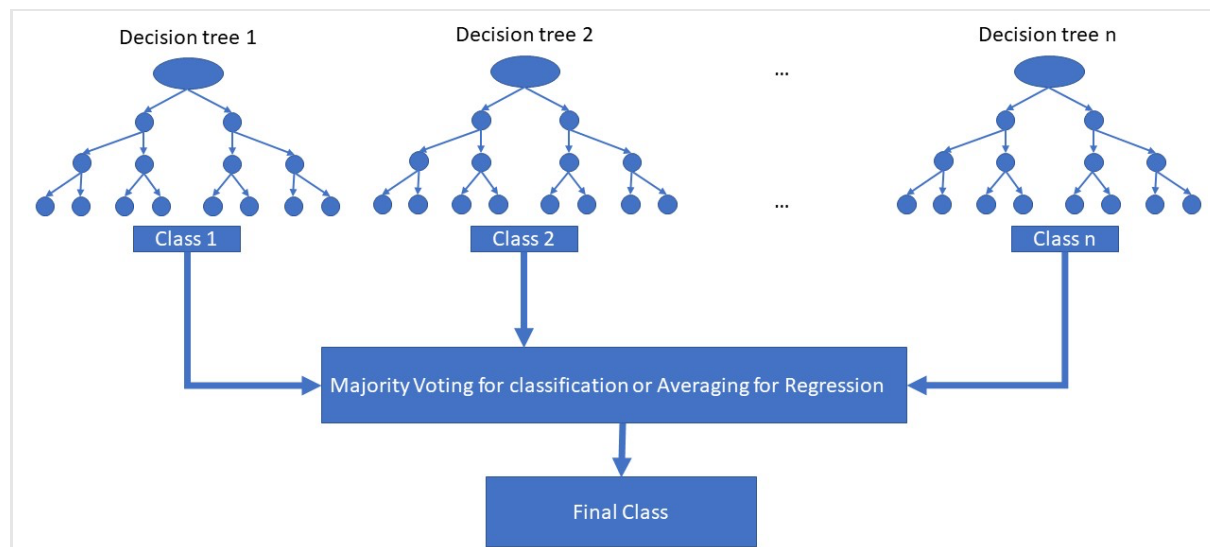


FIGURE 6: RANDOM FOREST ARCHITECTURE⁵

In terms of constructing a Random Forest, the algorithm follows a bagging (or bootstrapping) procedure where the so-called bootstrap samples are generated from the training dataset [18]. Each bootstrap sample has a fixed size and represents a specific subset of the training data. The data points included in each such subset are selected according to a uniform sampling strategy with replacement - i.e., each data sample can be reused and selected for training more than once. In addition to the bootstrapping procedure, Random Forests also utilize the so-called “feature bagging” method which ensures that the features for each individual decision tree in the ensemble are randomly sampled instead of using the complete feature set. This procedure aims at reducing the chance that individual features that have high predictive power will be universally chosen by large number of the decision trees and consequently the contributions of other, weaker features would be neglected [21]. With this in mind, after generating bootstrap samples according to the bootstrapping and the feature bagging procedures, every bootstrap sample is used for the training of an independent decision tree. The aggregate prediction of all such independent decision trees forms the final Random Forest output.

The main advantage of using the ensemble method as stated by [20] is that Random Forests show much higher accuracy when compared to a single Decision Tree. This is especially true for high dimensional data sets with large number of features. For such data sets a single Decision Tree will likely overfit the training set and would not generalize well for unknown instances. For lower-dimensional data, the Random Forest algorithm still performs reasonably well [20].

2.2.5 GRADIENT BOOSTED TREES AND XGBOOST

Gradient boosting is a machine learning technique predominantly used for both classification and regression tasks. Gradient boosting trains a single prediction model that consists in an ensemble of weak predictors. This means that each weak predictor typically has a low accuracy,

⁵ Based and adapted from <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>



D2.1: Accountability and Resilience Analysis

slightly better than random, but the combination of the weak predictors into the gradient boosting prediction model provides a high accuracy. Decision trees are the most used types of weak predictor, leading to a prediction model that is called a *gradient-boosted tree* model.

A trained gradient-boosted tree model is similar to a random forest model in that it is composed of many weak decision tree predictors. One main difference, however, is that in gradient boosting decision trees are built in a sequential manner. The main idea is that each new decision tree considers, and tries to cope with, the errors and weaknesses of previously trained decision trees. The training of gradient-boosted trees is different compared to most traditional machine learning models such as Neural Networks, logistic regression or SVM. During training, gradient boosting does not only optimize the parameters of a fixed and pre-defined decision function, but it also optimizes the function itself, i.e., each added tree changes the decision function. This objective of gradient boosting is to find the best function that approximates the training data. Gradient boosting trees aim to train an ensemble of simple models while Neural Networks, logistic regression or SVM aim to train a single complex model.

Training a changing function, in addition to its parameters, introduces a lot of complexity to the optimization problem, which induces an increased training time. XGBoost tackles this issue, and it is one of the fastest implementations of gradient boosted trees. One major improvement of XGBoost is that when building the weak tree predictors and looking for potential splits to create a new branch in the current tree, it does not consider all features. XGBoost analyses the distribution of features across the data points in the considered leaf of the tree, and it only considers features and splits that bring a positive gain on the loss. This reduces the search space and the complexity of the optimization. Another strength of XGBoost comes from its ability to be parallelized and take advantage of hardware optimization.

Gradient-boosted trees and XGBoost have many hyperparameters to be tuned for efficient training. One first parameter is the number of estimators, defining how many weak decision tree predictors will compose the model. A second parameter is the maximum depth of the trees, setting an upper bound on how many branches and leaves can each weak predictor have. Both of these parameters control the complexity of the gradient-boosted tree model. A high number of estimators and a large maximum tree depth enable the model to fit more complex problems, but it also comes with the risk of overfitting, which hinders the generalizability of the model. The learning rate is a third parameter, which is common to many machine learning algorithms. It controls how quick the learning can happen and conditions the convergence of the model by controlling the multiplying factor to weight updates at each training step. Finally, the regularization terms, for $L1$ and $L2$ regularization respectively, control the scale of the weights and ensure they are kept small. They are meant to ensure the generalization of the XGBoost model and prevent overfitting like for any other ML model. We can note that most of these hyperparameters control the generalizability and prevent overfitting of gradient-boosted tree models. The tendency to overfit is the main weakness of gradient-boosted tree models. This can be prevented by selecting a low number of estimators, a low maximum tree depth and large regularization terms, especially for $L2$ regularization.

Gradient boosted tree models have many advantages that explain their popularity. They usually provide a high predictive accuracy that cannot be trumped. They are very flexible, offer many hyper-parameters to tune and can be optimized on different loss function. Gradient boosted tree models do not require data pre-processing, such as normalization or scaling, and they can



D2.1: Accountability and Resilience Analysis

handle missing and sparse data very well. Gradient boosted tree models are also good at dealing with unbalanced datasets, where some classes are over- or under-represented. On the downsides, they tend to overfit because they just keep on minimizing the training error. Gradient boosted tree models are computationally expensive, especially when faced with a large feature space. The numerous hyper-parameters influence a lot the behaviour of the trained model, and it can be difficult to find their right values. Hyperparameters tuning can be long and computationally expensive. Finally, these models are not interpretable by nature, which poses a challenge for explainability.

2.2.6 BAYESIAN NETWORKS

Bayesian Networks [22] are a traditional probabilistic graphical model that has been used in Machine Learning methods to not only deal with uncertainty and complexity, but also reason about causal probabilities for scenarios given some evidence.

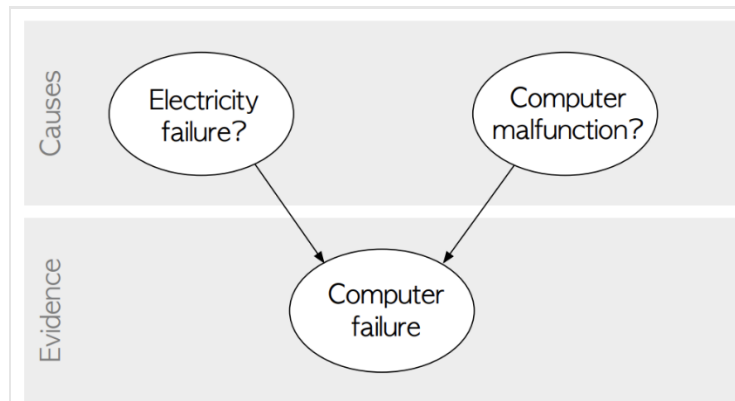


FIGURE 7: A DAG GRAPH REPRESENTING TWO INDEPENDENT CAUSES OF COMPUTER FAILURE [23]

A Bayesian network consists of two main parts: (1) a directed acyclic graph (DAG), which is a set of random variables represented by nodes and (2) a set of conditional probability distributions represented by directed edges. More specifically, the directed edge from a node A to a node B in the DAG graph shows that the variable A causes the variable B. We then define the conditional probability distribution of a node for every possible outcome of the preceding causal nodes. Considering an example in which the computer does not start correctly, we assume that there are two possible causes of computer failure: electricity failure and computer malfunction. Figure 7 depicts a DAG representing two independent causes of this failure.

Bayesian networks calculate the posterior conditional probability distribution of each of the possible causes given the observed evidence as follows:

$$P[\text{Cause} \mid \text{Evidence}] = P[\text{Evidence} \mid \text{Cause}] * P[\text{Cause}] / P[\text{Evidence}]$$

where $P[\text{Evidence} \mid \text{Cause}]$ represents the converse conditional probability distribution of the observing evidence given the cause. $P[\text{Cause}]$ and $P[\text{Evidence}]$ are the probability of the cause and the observing evidence, respectively. The joint probability distribution of all random variables in the graph factorises into a series of conditional probability distributions of random variables given their parents [22]. Concretely, given a list of random variables X_1, \dots, X_n , and



D2.1: Accountability and Resilience Analysis

parents (X) being the parents of the node X , the joint distribution for X_1 through X_n is calculated as $P(X_1, \dots, X_n) = P(X_i | \text{parents}(X_i))$, for $i = 1$ to n .

Bayesian Networks are widely used for modelling knowledge in various domains with uncertain knowledge, like image processing, medicine, data classification, etc. Recent research works apply Bayesian Networks methods in structure learning and classification. Several main advantages of Bayesian Networks are the ability to quantify the uncertainty in the parameters through posterior probability distributions and the ability to incrementally update the model. However, specifying prior knowledge in practice is difficult as we may need to consider concrete values for all parameters in our real model.

2.3 EXPLAINABLE AI METHODS

After presenting the six relevant ML algorithms, we will now discuss multiple XAI methods that can be used to achieve the explainability of the discussed black-box models. Thereby, we will present methods that allow local post-hoc explanations of individual model predictions as well as methods that facilitate global explanations of ML models.

2.3.1 LIME

LIME [24] is a widely popular technique used in interpreting outputs of black-box models in several fields and applications. LIME stands for Local Interpretable Model-agnostic Explanations. As the name suggests, LIME gives a *local* explanation, which means that it considers a subset of data when approximating explanations for model predictions. This technique is plausible under the premise that every complicated model performs linearly on a local scale. Nevertheless, LIME has recently gained high reputation due to its speed (relative to global explanation techniques) and convenience as it can interpret outputs irrespective of the type of black-box model (model-agnostic) which it wraps around.

A detailed description of the algorithm can be given as follows:

1. Generate a sample set of data points (also called perturbed data points) for a given input instance. This instance must be the one where you need an explanation for its output. The method of perturbing varies depending on the type of data (e.g.: tabular, text, images, etc.).
2. Map the perturbed data points to the original feature space so that they can be used as inputs to the black-box model.
3. Run the black box model on the perturbed inputs and generate corresponding predictions.
4. Weight the perturbed data points based on the distance to the original input instance.
5. From the perturbed data, choose a subset of features K that best characterize the black-box model outputs.
6. Train a simple interpretable model (e.g.: linear regression, decision tree, etc.) using the feature-reduced, weighted perturbed data.
7. Extract the feature weights from the simpler interpretable model and use them as explanations for the black box model's local behaviour.



LIME has caught the attention of the research community surrounding the field of cybersecurity. In literature [25] [26] authors have shown that ML based intrusion detection systems are widely capable of using LIME based explanations in attack detection. Authors of [25] have shown that, security information and event management systems has the potential to leverage the explanations generated by LIME to improve auto detection of alarm labels.

2.3.2 SHAP

SHAP (Shapley Additive Explanations) is an XAI technique that identifies the importance of each feature value in a certain prediction. For explaining individual predictions, it uses a concept called Shapley values. These Shapley values are a popular cooperative game theory technique that is based on the question of distribute a reward fairly among players of a group. Since the contribution of players for winning could be different, the reward should also be based on it. This concept is applied in order to explain AI predictions and to identify how features are contributing different amount to the final prediction. For this, Shapley values are used to calculate the contribution of each feature to the prediction by determining its marginal contribution for each possible set of features.

The formula for SHAP model explanation is given as:

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j$$

for the explanation model g . Here, SHAP specifies the explanation as a summation of shapley values $\phi_j \in \mathbb{R}$ for each coalition vector z' for a maximum coalition size M . Here, the coalition vectors are simplified features [27] of the set of features available for the model. For example, in image data, individual pixels can be simplified to produce a subset of pixels that make a coalition vector. The value obtained is finally added with the shapley value ϕ_0 where features are absent. Therefore, the steps for SHAP model calculations can be given as:

1. For each feature, calculate the coalitions set over the features in the model
2. For each coalition from $j = 1$ to M , calculate the shapley value and get their product
3. Get the summation of all the products of coalition and shapley values
4. Calculate the ϕ_0 and add to the total sum to get the explanation model for a simplified feature input z'

SHAP explanations is widely adopted as an XAI technique for AI applications due to its capability of identifying the most contributing features. These features would be of importance in the cybersecurity domain, as suggested by the work in [25] where the authors use SHAP to explain the most influencing features that contribute for prediction of cyber-attacks in an intrusion detection system using deep neural networks. Similarly, in [26], SHAP is used to identify most dominant features that influence anomalies generated by a ML model. Another example is in the context of IoT security [28], where SHAP is used for an intrusion detection system to evaluate two different ML models that use IoT datasets.



2.3.3 COUNTERFACTUAL EXPLANATIONS

The main idea of counterfactual explanations is based on the so-called counterfactuals. A counterfactual is a hypothetical scenario that illustrates how by perturbing the input features, we can force the model to generate a different prediction. A very common example [29][30][31][32][33] for a counterfactual explanation is a person applying for a loan. This situation can be framed as a binary classification problem (i.e., loan is approved or rejected) and the decision about the loan is made a ML/DL algorithm. If the loan is rejected, the bank typically provides a reason such as “bad credit history”, “unstable employment”, “missing paperwork”, etc. [29]. Such reasons do not provide any actionable steps for changing the decision of the model [29]. In contrast, the main purpose of counterfactual explanations is to address this problem by examining how small feature perturbations can change the model decision (e.g. approving the loan application). This suggests concrete actions that the user can take. For instance, in the loan application example, a counterfactual explanation could be “if you had a \$10,000 higher income, you would have been approved for the loan” [29][30][31][32]. The idea is that if the income input feature is perturbed and increased with \$10,000, the ML algorithm would change its prediction to the desired input. This example illustrates the standard structure behind most counterfactual explanations, which is summarized by C. Molnar as answering the question “*How would the prediction have been if input X had been different*” [32].

With this in mind, as pointed out by S. Wachter et al. [32], counterfactual explanations differ from the traditional definition of the term “explanation” in the XAI literature, which focuses on what the algorithm does internally in order to generate a given prediction [32]. In comparison, counterfactual explanations focus more on the contrast between the features that led to the current prediction and another set of slightly perturbed features that led to an alternative prediction [31][32]. As suggested by C. Molnar [31], the difference/contrast between these two sets of features can serve as a “human-friendly” explanation about “why” the model has made its decision. The reason is that counterfactual explanations narrow down the focus on only a select few features (e.g., only your annual income) instead of trying to address the relationship between all feature values and their corresponding labels (e.g. credit score, employment status, etc.) [31]. Due to this selective nature counterfactual explanations are easier to understand than complete explanations because they show how only a few causes led to a certain outcome [31]. Nevertheless, one challenge with counterfactual explanations is that usually there exist more than one counterfactual (also known as the “Rashomon effect”) [31] [33]. It is possible that from the multiple counterfactuals some might even contradict each other even though they lead to the same final outcome [31] [33]. For instance, one counterfactual might suggest to increase your income with \$10,000 and another one might suggest to not change your income, but instead to find a stable job and improve your credit score. Both of these would lead to an approved loan but they require completely different actions.

In this context, in order to assess if a counterfactual explanation is reasonable or not, there is a need for a criterion to do so. Such criteria is defined by C. Molnar [31] as:

1. The counterfactual instance has to be as close as possible to the desired, predefined prediction
2. The counterfactual instance should be as close as possible to the original instance and should include as few feature perturbations as possible.
3. There is a need for multiple alternative counterfactual explanations



D2.1: Accountability and Resilience Analysis

4. The counterfactual instance needs to have realistic feature values, which are possible to achieve in real life.

With this in mind, when it comes to generating counterfactual explanations, there are multiple different ways to do so. The naïve solution would be to generate these explanations by trial and error in a “brute-force” manner [33]. This is extremely impractical and therefore, there are multiple alternative algorithms proposed in literature which aim at generating counterfactuals in a more efficient way. Two examples for such more efficient methods are presented by Wachter et al. [32] and Dandl et al. [34].

Finally, what makes counterfactual explanations a very appealing approach in the context of XAI is that they provide a clear, human-understandable explanation about the model decision without requiring any knowledge about the model internals or the input data set [31]. In other words, instead of dealing with the complex “black-box” nature of ML and DL models, counterfactual explanations leverage it for their advantage. The main downside of the approach is the previously mentioned “Rashomon effect” [31] and the fact that they provide only local explanations. However, given the relatively low implementation effort of counterfactual explanations [31], they could still be extremely useful.

2.3.4 PERMUTATION FEATURE IMPORTANCE

Permutation feature importance is a global XAI method that measures the increase in the prediction error of the model after we permute the feature’s values across various data samples. To assess how important a specific feature is, we compare the initial model with the new model on which the feature’s values are randomly shuffled [19]. In other words, the effect of a feature is removed through a random reshuffling of the data to get new data, on which we calculate the prediction. This is different from another approach that simply retrains the model without this feature. A feature is important if shuffling its values increases or decreases the model error, because the model relies on the feature for the prediction in this case. Otherwise, a feature is classified as unimportant if permuting its feature’s values leaves the model error unchanged. Considering an example from [35], we develop an AI model using a person's height at age 10 to predict a person's height when she/he becomes 20 years old. As shown in the Figure 8 below, we shuffle data in a single column “Height at age 10 (cm)” and observe the results to determine if our model relied on this feature for predictions.

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24




FIGURE 8: SHUFFLING THE VALUES OF HEIGHT AT AGE 10 FOR PERMUTATION IMPORTANCE CALCULATION [35]



D2.1: Accountability and Resilience Analysis

The detailed algorithm [36] of Permutation feature importance is as follows:

Input: Trained model f , feature matrix X , target vector y , error measure $L(y, f)$.

- Estimate the original model error $e_{orig} = L(y, f(X))$ (e.g. mean squared error)
- For each feature $j \in \{1, \dots, p\}$ do:
 - Generate feature matrix X_{perm} by randomly shuffling the data of feature j (i.e., height at age 10) in the predictor while keeping the values of other features constant
 - Estimate error $e_{perm} = L(Y, f(X_{perm}))$ based on the predictions of the permuted data
 - Compute the feature importance score by calculating the decrease in the quality of the new predictions relative to the original ones as quotient $FI_j = e_{perm} / e_{orig}$ or difference $FI_j = e_{perm} - e_{orig}$.

Output: Once feature importance scores are computed for all features, we can rank them in terms of predictive usefulness.

To accelerate the computation without great loss of effectiveness, Fisher et al. also suggests splitting the dataset in half and swap the values of feature j of the two halves instead of permuting feature j [36].

Overall permutation feature importance provides a global insight into the model's behaviour. It does not require retraining the model and automatically takes into account all interactions between the feature under test with other features. On the other hand, we do need to have the true outcome, in order to precisely measure the feature importance scores. Also, the results of this method may vary greatly due to the randomness of the process of shuffling the feature.

2.3.5 PARTIAL DEPENDENCE PLOT

Partial dependence plot (PDP) [37] is a global XAI method that allows to visualise and analyse interaction between the prediction and a set of input features of interest. Like permutation feature importance in the previous section, PDP is calculated after a model has been fit on real data. While permutation feature importance explains the AI models by showing what variables most affect the outcome, PDP focuses on how a specific feature affects model predictions. If we select only one feature, we will draw a 2D plot. In case of having two features of interest, a 3D plot will be built as the output of this XAI method. The algorithm of PDP can be summarised at a high level as follows:

Input: A trained model and a set of features of the training data

- We select a single feature from the feature set and then use the trained model to predict the outcome of that feature of interest.
- We repeatedly alter the value for that feature to make a series of outcomes. For example, in the example of prediction of height at age 20 discussed above, we could obtain the prediction for different values of the height at age 10 in cm: 130, 142, 147, etc.
- We build the plot with the value change in the selected feature on the X-axis and the change of the outcome on the Y-axis



Output: A partial dependence plot representing the impact of a feature of interest towards model prediction.

Considering the California housing dataset [38], we develop an ML model for predicting house prices in any district in California, given information regarding the house in the districts (house age, number of rooms, number of bedrooms), the demography (population, income, house occupancy) and the location of the districts (latitude, longitude). As shown in the Figure 9 below, we can draw two 2D plots showing the effect of two features, namely the average occupancy “AveOccup” and the house age “HouseAge”, on the median house price. Clearly, we observe a linear relationship between the house price and the average occupancy, especially when it is less than 3 people. Furthermore, we can also draw a 3D plot showing the relationship between the house price and joint values of those two features.

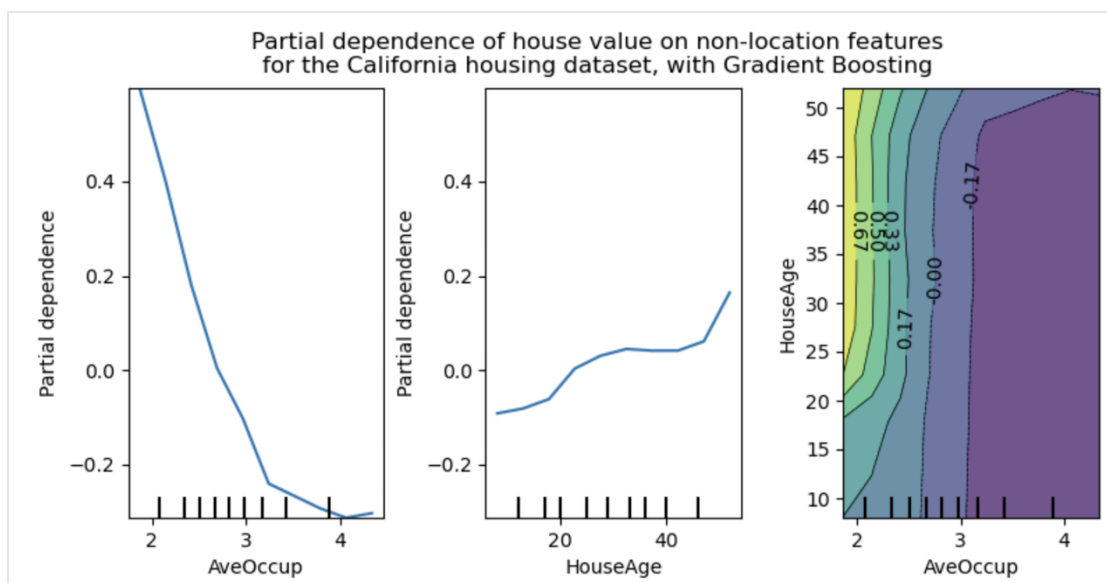


FIGURE 9: PARTIAL DEPENDENCE 2D/3D PLOTS FOR THE CALIFORNIA HOUSING DATASET [38]

In general, the computation of PDP is intuitive and the interpretation is clear if the feature of interest is not correlated with the other features. However, the biggest issue with PDP is the assumption of independence which is often violated in practice. Another disadvantage of PDP is that the realistic maximum number of features in a partial dependence function is two, as humans can’t process three- or higher dimensional plots.

2.3.6 T-SNE

A common challenge in the AI domain is the analysis of high dimensional data, which often times is hard to visualise in a human-understandable way. One popular algorithm that addresses this challenge is t-distributed stochastic neighbour embedding (t-SNE). t-SNE is an unsupervised dimensionality reduction technique that aims at finding an accurate low-dimensional representation of high-dimensional data points. Typically, the low-dimensional representation is generated in a 2D or 3D space [39]. This could allow the ML practitioner to explore the high-dimensional data and its arrangement in a more visually comprehensive manner, which would be otherwise impossible in a space with more than three dimensions. One



D2.1: Accountability and Resilience Analysis

practical example for using t-SNE for visualisation purposes is presented by L. Maaten and G. Hinton [39] in Figure 10. The figure demonstrates how data samples from the popular MNIST dataset can be visualised in a 2D plane. The MNIST dataset contains handwritten digits represented as 28 by 28 pixel images (i.e. 784 dimensions). By applying t-SNE, the authors managed to map the high-dimensional data into an equivalent 2D representation, where each number has its own cluster [39]. This makes the visualisation and consequently the data analysis and exploration more straightforward.

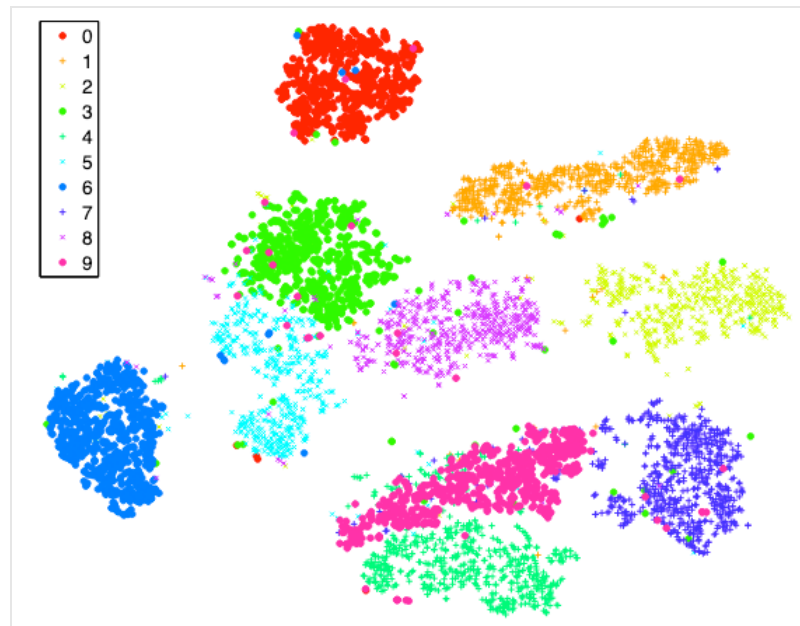


FIGURE 10: T-SNE APPLIED ON THE MNIST DATASET [39]

In terms of the inner workings, t-SNE is based on the Stochastic Neighbor Embedding (SNE) [40] algorithm, but improves upon it by using an alternative cost function and a t-distribution instead of Gaussian when computing the point similarity in the low-dimensional space [39]. The algorithm follows a multistep process that can be summarised as follows [39]:

1. First, t-SNE measures Euclidean distance between the high dimensional data points and converts these into conditional probabilities in the form $P(x_j|x_i)$. This illustrates the probability that x_i would pick x_j as its neighbour, if neighbours were determined based on their probability density value under a Gaussian distribution centered at x_i [39]. In other words, data points with small Euclidean distance would also have a high probability to be picked as neighbors [39].
2. Afterwards, t-SNE maps the high dimensional representations of all pairs x_i and x_j into their lower dimensional counterparts y_i and y_j [39] and computes the low dimensional similarities between these points in a similar manner as in step 1. However, as mentioned above, in the low dimensional space, t-SNE utilises a student's t-distribution instead of Gaussian to compute the similarity between the points [39]. The conditional probability in the low dimensional space is denoted $Q(y_j|y_i)$.



D2.1: Accountability and Resilience Analysis

3. Finally, t-SNE tries to find a low dimensional representation that minimizes the difference between $P(x_j|x_i)$ and $Q(y_j|y_i)$ [39]. This is achieved with the help of a gradient descent based on Kullback-Liebler divergence [39], which compares the difference between $P(x_j|x_i)$ and $Q(y_j|y_i)$ for all data samples and re-arranges these into identical clusters as the ones previously observed in the high dimensional data representation.

In the context of XAI, t-SNE can be utilized successfully as a pre-modelling explainability technique. As mentioned previously, one of the main advantages of t-SNE is making high-dimensional data human-understandable by visualizing it in a lower two- or three-dimensional space. This could be particularly useful during the exploratory data analysis step of many Machine Learning pipelines. Additionally, as demonstrated by Karpathy [42], in specific use cases t-SNE can also be used to examine and validate the behaviour of ML models.

Despite being very useful for visualising high-dimensional data, sometimes t-SNE might generate low-dimensional representations that can be misinterpreted and even misleading [41]. Additionally, t-SNE is fairly slow when applied on large datasets and requires hyper-parameter optimisation, in order to achieve decent results.

2.3.7 LAYER-WISE RELEVANCE PROPAGATION

In recent years the Computer Vision domain has gained a lot of popularity and was subject to a lot of technological advancements. However, one challenge that remains unsolved is the explainability of the model predictions. The main reason is that most ML models perform their tasks (e.g., classify images, detect objects, etc.) in a black-box manner. Layer-wise relevance propagation (LRP) is an XAI technique that addresses this challenge. In particular, LRP provides the ML practitioner with insights about the model decision by visualizing the individual feature values that contributed most for the generated prediction.

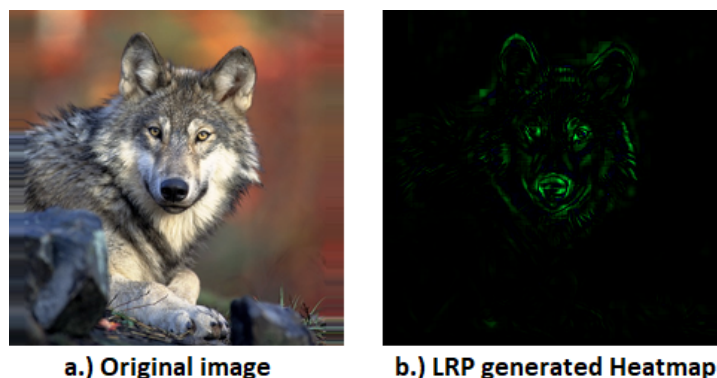


FIGURE 11: LRP EXAMPLE (GENERATED WITH [HTTPS://LRPSEVER.HHI.FRAUNHOFER.DE/IMAGE-CLASSIFICATION](https://lrpserver.hhi.fraunhofer.de/image-classification))

This idea is illustrated in Figure 11, where a classifier predicts that the original input image contains a “timber wolf”. By applying it on the classifier for this particular input, LRP generates a heatmap where pixels of particular significance for the model prediction are marked with more intense colours. By examining the original image and the corresponding heatmap generated by LRP, the ML practitioner or a domain expert could assess if the model's decision



D2.1: Accountability and Resilience Analysis

is reasonable and supported by the correct patterns in the input features [43], instead of additional external factors (e.g. “Clever Hans” behaviour [44]).

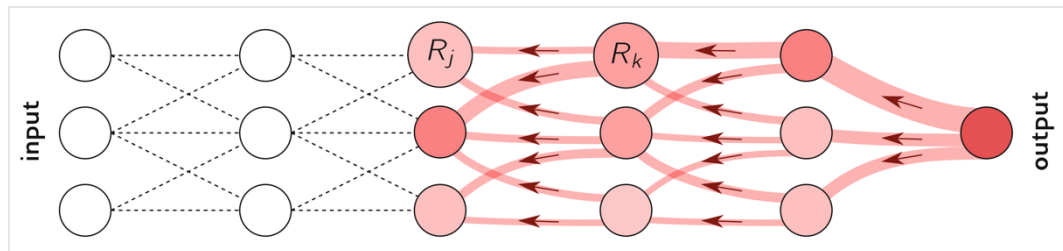


FIGURE 12: BACKPROPAGATION OF RELEVANCE FROM THE OUTPUT TO THE INPUT LAYER (TAKEN FROM [43])

When it comes to its inner workings, LRP is based on the idea of propagating the so-called “relevance scores” from the output layer through the hidden layers back to the input layer [43]. This process is visualised in Figure 12 and in Figure 13, where the relevance propagation from the network’s prediction back to the input features is demonstrated. In both figures more intense colours indicate higher relevance score. The relevance score of each neuron is computed with the help of the so-called “propagation rules” [43]. The most basic rule is denoted by G. Monavon et al. as “LRP-0” and has the following formula [45]:

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_0^j a_j w_{jk}} R_k$$

In this formula, j and k represent neurons from two consecutive layers in the neural network, where j is the neuron in the lower layer and k the neuron in the higher layer [43]. The concept behind this rule is fairly intuitive – the numerator expresses the contribution of neuron j to neuron k computed as the neuron activation multiplied by the corresponding weight value [43]. The result is then divided by the sum of all neuron contributions from the lower layer (i.e. the denominator), which enforces the so-called “conservation property” [43] that states that the whole amount of relevance value received by a neuron has to be redistributed in the exact same amount to the lower layer [43] - i.e. no loss of relevance value is allowed to occur during the backpropagation through the network layers. In the LRP-0 formula above, since the neuron j contributes to multiple neurons in the next layer, its relevance score is computed as the sum of its contributions to all neurons in this next layer, which is expressed as the outer sum in the LRP-0 equation above [43]. By applying the LRP rules from the output towards the input layers, the relevance scores are propagated and can be used to build a heatmap similar to the one represented in Figure 13. The upper part of this figure depicts a prediction for the class “cat” that is obtained by forward-propagation of the pixel values $\{x_p\}$, and is then encoded by the output neuron x_f . In the lower part of this figure, the output neuron is assigned a relevance score $R_f = x_f$ representing the total evidence for the class “cat”. Relevance is then backpropagated from the top layer down to the input, where $\{R_p\}$ denotes the pixel-wise relevance scores, that can be visualized as a heatmap.

Such visual representations are extremely useful for explainability purposes. However, as discussed by G. Montavon et al. [43], despite being fairly intuitive, the universal application of LRP-0 rule across all neurons in the network has its flaws and the usage of more robust propagation rules could be beneficial. Such alternative rules are presented by the authors in



D2.1: Accountability and Resilience Analysis

[43]. Important to note here is that LRP is flexible and allows using different propagation rules in the different network layers [43]. Choosing a propagation rule with the optimal parameters depends on the explanation quality provided by LRP [43]. In that context, G. Montavon et al. suggest the evaluation of LRP explanations quality with regards to two main XAI properties - fidelity [46] and understandability [43]. More specifically, the ML practitioner using LRP should strive to find and use parameters and propagation rules that maximise the fidelity and understandability of the generated explanations.

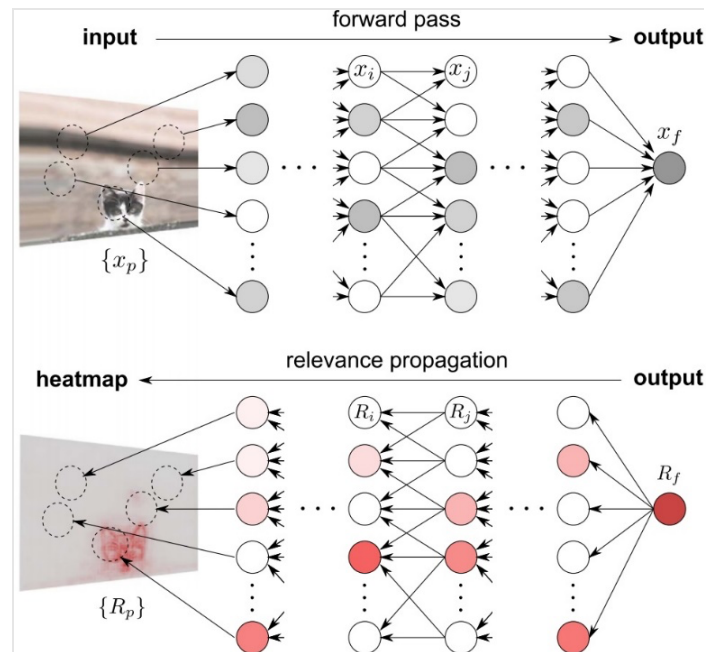


FIGURE 13: LRP METHOD VISUALISED (TAKEN WITHOUT CHANGES FROM [45])

With this in mind, LRP is a flexible XAI technique that delivers human-understandable explanations. Due to the wide variety of existing propagation rules, it can be applied to multitude of ML models [43] and can be implemented efficiently with the help of current SOTA libraries [43].

2.3.8 OCCLUSION SENSITIVITY

Occlusion sensitivity is an explainability method that is agnostic to the underlying model that is used for classification. To generate explanations based on occlusion sensitivity, the input data of a Convolutional Neural Network is systematically occluded with a grey mask, with which the prediction of a trained CNN model is estimated for variation in classification score on the basis of an initial prediction of the CNN model on original input data (i.e., unoccluded image). Considering a trained model $f: \mathbb{R} \rightarrow \mathbb{R}$ that takes input $X = (x_1, \dots, x_d)$, a real value vector of features, and outputs a score. The output score is compared to the defined threshold for the classification decision. To understand the features that impact the prediction of the output of the model $f(x)$ for a specific input, a local linear approximation of the decision function $f(x) \approx \sum_{i=1}^d \underbrace{[\nabla f(\tilde{x})]_i}_{H_i} * (x_i - \tilde{x}_i)$ where x_i is a reference point, can be used [48]. The contribution of

feature i to the prediction is denoted by summand H_i . A feature x_i is strongly relevant if it differs



D2.1: Accountability and Resilience Analysis

from the reference value x_i , and the model output should be sensitive to the presence of that feature, i.e., $[\nabla f(\tilde{x})]_i$. An explanation for the prediction can then be formed by the vector of relevance scores R_i . It can be given to the user as a histogram over the input features or as a heatmap.

A summary of the algorithm is described below (graphical flow is also presented in Figure 14). Using occlusion explainability method, input x_i is perturbed using a patch $mask_i$, where $mask_i$ represents the indicator vector for the patch and “•” denotes the element-wise product. The difference of the score of the occluded output and the score of the original output is compared as $H_i = f(x) - f(x \bullet (1 - mask_i))$ to understand how perturbation affected the function. $(H_i)_i$ represent the location where the occlusion has caused the strongest decrease of the function and can be used to build a heatmap for visualization.

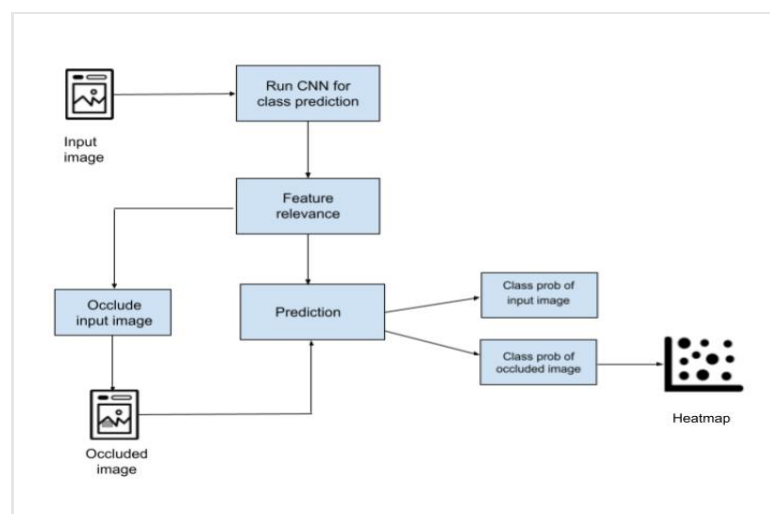


FIGURE 14: GRAPHICAL PRODUCE OF OCCLUSION SENSITIVITY METHOD (AUTHORS' OWN CONTRIBUTION)

Occlusion procedure algorithm

Input: Data image to occlude with occlusion mask, Image index by X_i

- Estimate the shape of the image (X_i)
- Estimate length of the image (X_i)
- Produce occluded version $X_{occ} \leftarrow \text{Copy}(X)$
- Select occluding value mask
- Estimate area to be occluded ($w_{size} * w_{idx}$)
- Assign occluded value to the area

Output: Occluded image

Occlusion sensitivity have gained importance for rapid localization of critical features from images. By applying diffusion weighted magnetic resonance imaging (DWMRI) method is possible to identify radial diffusivity information from data of patients that have parkison diseases [47]. To explain further the intuition behind the localization of critical features, we rely on an IoT based artificial intelligence model for identification of nature objects (see Figure 15). Here, occlusion sensitivity was performed by hierarchically masking varying portion of the



D2.1: Accountability and Resilience Analysis

input images from DWMRI. The hierarchical occlusion sensitivity approach was found to localise important features that influences prediction of the model 20 times more than baseline techniques used and provided opportunity for faster explanation of model's intuition.

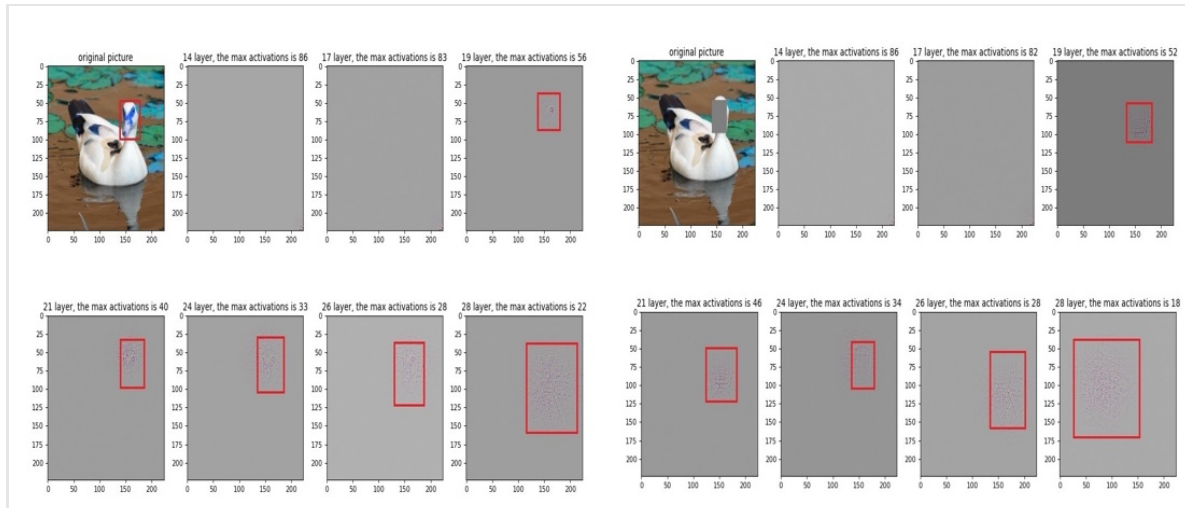


FIGURE 15: OCCLUSION SENSITIVITY FOR INTERPRETABILITY FOR MEDICAL IMAGING IN IOT APPLICATIONS (AUTHORS' OWN CONTRIBUTION USING THE TOOL [118]⁶)

2.3.9 CAM AND GRAD CAM

Class Activation Map (CAM) and Grad CAM are local, post-hoc XAI methods. They use the back-propagation mechanism and explain Convolutional Neural Networks, a deep learning algorithm for imagery data. For CNN, this means propagating backward from the last layer to find the corresponding features in the image causing the output and highlighting them to provide the explanations [50].

CAM: Conventionally, in CNN, the end of the network structure consists of a fully connected layer, followed by a softmax layer [51]. CAM uses a particular type of convolutional neural network, where the fully connected layer is replaced by a global average pooling (GAP) layer (Figure 16). It was observed that using the GAP layer in this way helps to avoid overfitting and was used to regularize training data [51]. For this particular type of CNN, CAM proposes that the weighted average of the feature map from the last layer convolutional layer generates the localization map of features causing the output [49]. Figure 16 shows the working of CAM; the model identifies an Australian terrier in the image, performs the weighted average of the feature map by using the weights of the softmax layer, and highlights discriminative parts of the image causing this output. The outputs are normalized for visualization [49] [52] [53].

If F^k is the output of GAP layer, and S_c is the output of the softmax layer for class c , mathematically they can be represented as [49]:

⁶ The original input image can be found here: [https://hawaiibirdingtrails.hawaii.gov/wp-content/uploads/Muscovy-Duck-female Michelle-Moore-1024x1024.jpg](https://hawaiibirdingtrails.hawaii.gov/wp-content/uploads/Muscovy-Duck-female_Michelle-Moore-1024x1024.jpg)



D2.1: Accountability and Resilience Analysis

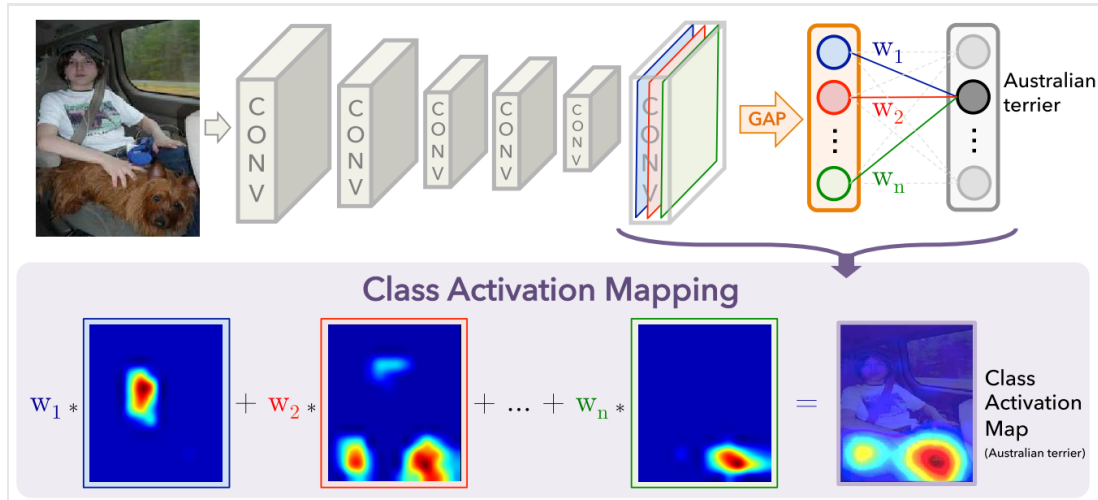


FIGURE 16: CLASS ACTIVATION MAPPING: CAM HIGHLIGHTING THE DISCRIMINATIVE IMAGE REGIONS USED BY CNN FOR CLASSIFYING THE AUSTRALIAN TERRIER [49]

$$F^k = \sum_{x,y} f_k(x, y),$$

where $f_k(x, y)$ is the activation of unit k in the last convolutional layer at the location (x, y)

$$S_c = \sum_k w_k^c F^k,$$

where w_k^c are the weights of F^k for class c . Replacing the value of F^k in the last equation we have,

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_k \sum_{x,y} w_k^c f_k(x, y)$$

From the definition, the class activation map M_c for class c at each spatial element can be defined as:

$$M_c(x, y) = \sum_{x,y} w_k^c f_k(x, y)$$

Thus, essentially the output of softmax layer $S_c = \sum_{x,y} M_c(x, y)$, and hence $M_c(x, y)$, the class activation map directly indicates the importance of the activation at (x, y) causing the output as class c .

Grad-CAM: In contrast to CAM, which applies to a particular CNN network with GAP layer, Grad-CAM can be applied for CNN with fully connected layers. It generalizes CAM by taking the gradient of the scores for a given class w.r.t. the feature map activations and then performs the global average pooling (equation 1 below). After this, it performs the weighted average using these new weights, which is similar to CAM. In Grad CAM, an additional ReLU layer is added after this to generate the feature visualizations (equation 2 below). Thus, the steps before ReLU provide a generalization of the CAM algorithm to be applied to other fully



D2.1: Accountability and Resilience Analysis

connected CNN networks. Mathematically the new gradient α_k^c , and the Grad-CAM heat map $L_{Grad-CAM}^c$ can be defined by [52] [53]:

$$(eq. 1) \quad \alpha_k^c = \frac{1}{Z} \sum_{x,y} \frac{\partial S_c}{\partial f_k(x,y)}$$

$$(eq. 2) \quad L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c f_k(x, y))$$



3 ACCOUNTABILITY AND RESILIENCE ANALYSIS OF IDENTIFIED AI ALGORITHMS

After having discussed the required technical and theoretical foundations, the following section constitutes the main section of this deliverable document. More specifically, we will analyse the identified algorithms with respect to their accountability and resilience characteristics in the following section. We will begin the section by analysing the explainability (since we see the explainability of ML algorithms as one of the means to achieve accountability in SPATIAL) of the algorithms and identifying applicable XAI methods. Furthermore, to characterize the resilience of the algorithms, we will investigate the six ML algorithms with respect to their resilience to adversarial ML attacks. Finally, we review the four SPATIAL use cases and illustrate their need for accountable and resilient ML algorithms.

3.1 ANALYSIS OF IDENTIFIED AI ALGORITHMS

In the following section, the identified algorithms are analysed with respect to their accountability and resilience. Thereby, we explore the explainability of ML algorithms, in order to understand their accountability. This is justified by the fact that we see the explainability of ML algorithms as a means to achieve accountability in SPATIAL. Therefore, algorithms are examined concerning their intrinsic explainability as well as the explainability enabled through state-of-the-art XAI methods. An overview of the identified applicable XAI methods for each discussed algorithm is illustrated in Table 1, where a “tick” means a certain XAI method can (while a “cross” means cannot) be used for a corresponding ML algorithm.

Furthermore, the ML algorithms are studied in terms of their vulnerability to adversarial attacks (e.g. poisoning attacks, evasion attacks, data inference attacks, and model stealing attacks) to explore their resilience. Specifically, we will identify literature (see Table 2) that has studied the vulnerability of specific ML algorithms against these attacks. To limit the scope, we will distinguish between data poisoning attacks, evasion attacks, data inference attacks, and model stealing attacks. In this context, data poisoning attacks aim to compromise ML models at training time by supplying poisoned training data. Here, the attacker aims to degrade the trained model's prediction performance (e.g., in terms of accuracy) [116] or implant backdoors [116]. In contrast, evasion attacks, data inference attacks, and model stealing attacks target to compromise models at inference time. More precisely, an attacker aims to use evasion attacks to generate adversarial examples. The latter are minimal perturbations of the input that result in an altered and invalid output prediction of the attacked model during operation [117]. Thus, the intended behaviour of deployed ML models can be selectively altered by the attacker. On the other hand, data inference attacks try to attack deployed ML models to obtain information about the used training data [117], which can lead to serious privacy issues. Alternatively, model stealing attacks can result in a violation of intellectual property rights. Here, an attacker attempts to mimic the behaviour of an ML model by approximating the model and its internal parameters based on obtained input-output pairings [117].



D2.1: Accountability and Resilience Analysis

TABLE 1: OVERVIEW OF THE APPLICABLE XAI METHODS OF THE IDENTIFIED ML ALGORITHMS

Algorithm XAI Method	DNNs	SVMs	Decision Trees	Random Forests	GBTs & XGBOOST	Bayesian Networks
LIME	✓	✓	✓	✓	✓	✓
SHAP	✓	✓	✓	✓	✓	✓
Counterfactual Explanations	✓	✓	✓	✓	✓	✓
Permutation Feature Importance	✓	✓	✓	✓	✓	✓
Partial Dependence Plots	✓	✓	✓	✓	✓	✓
t-SNE	✓	✓	✓	✓	✓	✓
LRP	✓	✗	✗	✗	✗	✗
Occlusion Sensitivity	✓	✗	✗	✗	✗	✗
CAM and Grad CAM	✓	✗	✗	✗	✗	✗

3.1.1 (DEEP) NEURAL NETWORKS

3.1.1.1 Accountability and Explainability

Neural networks cannot be considered as transparent machine learning models by default. Even the simpler forms of neural networks such as multi-layer perceptrons are not inherently interpretable for a human. With increasing depth of neural networks' layers, the black-box nature of the model increases, drifting the model further away from its inherent interpretability. Each additional neuron layer adds a set of weights and activation functions that contributes to the final output of the model. The raw arrangement of these weights is not conceivable directly by the end user and thus they are not useful in interpreting how the model has identified the input attributes and generated the output. This results in an overall weak explainability and thus only limited accountability. However, as depicted in Table 1, a range of XAI tools can be used to achieve the explainability of neural networks on local and global level. Saliency based methods such as SHAP and proxy-based methods such as LIME can be used to interpret the



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

model locally. SHAP is even capable of giving global interpretation of neural network models. Counterfactual explainers are another great example of local interpreters that can be applied to DNNs. LRP, occlusion sensitivity, and CAM/GradCAM are specifically designed DNN interpretation techniques that are used to understand important areas of an input especially when image data is involved. Furthermore, permutation feature importance and PDP techniques are also applicable to neural networks to gain global level explainability.

3.1.1.2 Resilience

Neural networks have a complex architecture by design, and as a result, they are vulnerable to many attack vectors. Identifying the effect of an attack on a neural network is even harder to localize due to their large parameter space. From data collection and training to deployment and inference, neural networks are open to adversarial attacks. Basically, the intention of training a neural network is finding a delicate balance between generalization and discrimination of data over the targets. Sometimes a simple fault-injection in the model can cause catastrophic misclassifications depending on the applications.

Poisoning attacks occur when the training data/algorithm is manipulated by an attacker with malevolent intents. Poisoning attacks on NNs are heavily studied in current literature. In a clean-label poisoning attack [58][59][60][62] even the expert data labellers would fail to identify the poisoned inputs. These attacks are also capable of creating backdoors in neural networks [60]. The NN model itself can also be poisoned if the attacker has access to model parameters [61].

Attackers can commit evasion attacks on neural networks under black-box or white-box configurations using carefully crafted data samples embedded with imperceptible adversarial noise. Such attacks could either affect selected targets or decrease the general prediction accuracy. Adversarial samples are generated utilizing various elements of NNs such as the gradient [63] [64], model score [65] and decisions [66].

Already trained NNs which are not publicly available (or black-box) are open to model stealing attacks where the adversaries try to extract the model parameters which can be used to obtain similar results as the original model [67] [68]. Model stealing lays the groundwork towards generating adversarial examples.

Inference attacks are also a widely studied class of attacks when it comes to neural networks. Here the adversary's motive is to determine things such as whether a datapoint belongs to the original dataset (membership inference) [69], the dataset as a whole (model inversion) [70], or extra information learned by the model that are not related to the original task (attribute/property inference) [71].

The effect of some of the above-mentioned attacks can be mitigated with adversarial training and ensemble methods. Most of the current defences only prevent specific types of adversarial attacks where the need for more general prevention mechanisms is required more than ever [72].



TABLE 2: IDENTIFIED REFERENCES OF THE RESPECTIVE ADVERSARIAL ATTACKS AGAINST THE ML ALGORITHMS

Algorithm Attack Type	DNNs	SVMs	Decision Trees	Random Forests	GBTs & XGBOOST	Bayesian Networks
Poisoning Attacks	[58][59] [60][61] [62][72]	[73] [75]	[88][89]	[91][101] [106]	[91][105]	[110] [111]
Evasion Attacks	[63][64] [65][66]	[76][77] [78][79]	[89][90] [91][92] [93]	[89][91] [94]	[91][92] [94][95]	/
Model Stealing Attacks	[67][68]	[80][81] [82]	[96]	[98][102]	[103]	/
Data Interference Attacks	[69][70] [71]	[83][84] [85]	[97][98] [99][100]	[102]	[107]	/

3.1.2 SUPPORT VECTOR MACHINES

3.1.2.1 Accountability and Explainability

SVMs are widely used in networking and security due to its faster inferencing and its capability to classify non-linear numerical data. There are mainly two types of SVMs – namely linear SVMs and kernel trick SVMs. Linear SVMs are more interpretable on their own. Here, the weights of the model can be broken down to a product between input sample and weight vector. Therefore, the weights in linear SVMs are directly representative for the importance of the features identified by the model. On the other hand, SVMs that use kernel trick are more suitable to classify large non-linear datasets. SVMs achieve this by transforming these feature spaces into linearly separable higher dimensional spaces. With this transformation, the model loses some interpretability making the weights and parameters of the model losing its linear relationship with feature importance. SVMs that operate on two- or three-dimensional data provide a high level of explainability, since their decision boundary can be directly visualized in 2D or 3D plots. However, SVMs operating on high-dimensional data are losing this characteristic. In this case, dimensionality reductions could be used to visualize the decision boundary. However, these methods make it difficult to understand the relationship between original data features and model predictions. As an alternative, model-agnostic XAI methods such as LIME, SHAP, and counterfactual explanations can be used with SVMs to achieve local explainability (see Table 1).

3.1.2.2 Resilience



SPATIAL project is funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

SVMs are extensively proposed in tasks such as malware detection, intrusion detection and spam filtering. However, SVMs themselves are susceptible to many of the commonly seen attack types among ML models such as poisoning, evasion, model stealing, and data inference attacks. SVM with kernel trick (k-SVM) are also vulnerable to adversarial attacks just like the linear SVM. When it comes to real world application of SVM, it becomes imperative to reinforce the security of the model itself despite the type of application.

Dataset poisoning attacks are perceivable in SVMs during the training process or even when retraining on data collected after deploying it in an unsupervised or semi-supervised manner. If the adversary has access to feature samples and the training dataset, then it is quite possible to stage a label flipping attack or even create backdoors in the model. Even without the full training dataset, adversaries can perform limited knowledge poisoning on SVM models through surrogate datasets. Poisoning attacks on SVMs are extensively studied in the recent literature [73] [75].

Evasion attacks in SVMs are studied under perfect knowledge (white box) and limited knowledge (black box) conditions. As opposed to poisoning attacks, evasion attacks happen during the testing phase or deployment stage. In recent literature these attacks are studied with reference to SVMs under various applications including spam filtering, network intrusion detection, etc [76] [78]. Kernel trick SVMs are no exception and can be compromised through evasion attacks executed through gradient-descent [79]. In malware detection using SVMs, it has been shown that the effectiveness of evasion attacks can increase when it is combined with other attack types such as collusion [77].

Model stealing/extraction attacks on SVMs can reveal the exact decision boundary making them significantly dangerous [80]. Lowd-Meek attack in [81] is considered a successful method to steal linear SVM models. Similarly, kernel trick SVMs are also susceptible to model stealing attacks as shown in [82]. Membership inference attacks and attribute inference attacks on SVMs can be potentially used to violate the privacy of users when it comes to social media information such as shown in [83][84].

As means of protection against privacy related attacks, differential private SVMs are a popular proposal in related research [85]. Adversarial training of both linear and kernel SVM models is also commonly seen in literature to make trained models robust against evasion attacks.

3.1.3 DECISION TREES

3.1.3.1 Accountability and Explainability

A decision tree is an ML algorithm that offers high expressive power in modelling linear and non-linear relationships, while still providing intrinsic interpretability. As already described in Section 2.2.3, a decision tree performs a series of hierarchical decisions, each involving a single feature at a time that is tested against a threshold value. Each prediction from a decision tree can be explained by providing the path from the root to the prediction leaf, and each test involved in reaching that leaf. Thereby, each node in the path can be interpreted as a human understandable text description in the form of “*If ... then ... else*”. In other words, a decision tree resembles human reasoning by providing a transparent concatenation of human-



D2.1: Accountability and Resilience Analysis

understandable if-then-else explanations [86]. Thus, decision trees provide local explainability by enabling transparent reasoning of individual decisions.

Besides local explanations of individual decisions, a decision tree also provides global explainability due to its decomposability and algorithmic transparency, which enables a global analysis of the trained model [87]. During the training process, the decision tree generates transparent prediction rules, which represent the knowledge learned from the provided training data. These generalizing prediction rules render the decision-making process globally comprehensible. Furthermore, they allow for revealing the impact of individual data features on the decision process, which enables deeper insights and analyses.

In summary, a decision tree with its inherent interpretability provides both local explainability of individual decisions and global explainability of the decision tree model, resulting in a high degree of accountability. However, the high transparency comes at the cost of significantly reduced expressive power compared to the other black-box models discussed in this work (e.g. DNNs or random forests). In addition, we would like to mention that while many of the local and global XAI methods discussed in this paper can be applied to decision trees (see Table 1), it is generally not necessary due to their inherent interpretability [87]. Instead, decision trees are often used in XAI methods due to their intrinsic trade-off between interpretability and expressive power. For example, decision trees can be used as surrogate models to approximate opaque black-box models and thus enable the explainability of these.

3.1.3.2 Resilience

In recent years, several works have demonstrated the vulnerability of decision tree models to adversarial attacks, which raises serious security concerns regarding their resilience characteristics. For example, several authors report that data poisoning attacks can be used against decision tree models with high effectiveness. For instance, Mozaffari-Kermani et al. [88] draw attention to poisoning attacks in the domain of eHealth by describing a data injection attack that injects malicious samples into the training data. Their experiments analysed the accuracy of decision tree models when performing binary classification with varying injection rates. Thereby, they observed a significant accuracy drop of the attacked decision tree models, demonstrating the vulnerability of decision tree models against data injection attacks. In another example, Newaz et al. [89] aimed to comprise an ML-based smart healthcare system consisting of eight different smart medical devices by applying data poisoning attacks. In the conducted experiments, the authors poisoned the dataset with data modification methods and evaluated the accuracy drop. This work reports an accuracy drop of up to 27.31% in their experimental settings, which indicates again the vulnerability of decision trees to poisoning attacks.

In addition to poisoning attacks, various existing work demonstrated that decision trees are also vulnerable to evasion attacks. For example, besides poisoning attacks, Newaz et al [89] also investigated the vulnerability of their ML-based smart healthcare system against evasion attacks. More specifically, the authors were able to execute the HopSkipJump [90] attack in a white-box setting and successfully generated adversarial examples against a decision tree model, which suggests that decision tree models are vulnerable to this attack. Successful evasion attacks against decision tree models have also been demonstrated in the area of ML-based NIDSs. Alhajjar et al. [91] successfully applied perturbation methods based on generative



D2.1: Accountability and Resilience Analysis

adversarial networks against decision trees and showed that these are vulnerable to adversarial examples. Finally, Chen et al. [92] demonstrated the vulnerability of tree-based ML models to adversarial examples by successfully applying multiple evasion attacks, including Papernot's [93] L_∞ -versions of Kantchelian's [94]. However, the authors not only showed the vulnerability of tree-based models but also proposed an algorithm to construct more robust trees.

Model stealing attacks have been identified as another vulnerability of decision tree models. In [96], Tramer et al. attacked decision trees from the online service BigML⁷. They demonstrated that they were able to successfully extract model parameters of the underlying decision trees with relatively few input-output API queries. This manifests the vulnerability of decision trees against this kind of attack and emphasizes that the deployment context of decision tree models must be carefully chosen and strongly protected to provide confidentiality and protect intellectual properties.

Finally, some researchers have also demonstrated the vulnerability of decision trees against data inference attacks. In [97], Truex et al. demonstrated a successful membership inference attack (MIA) against decision trees in a black-box setting. In this context, it was shown that the effectiveness of the attack in terms of MIA accuracy depends strongly on the training dataset used and the attacker's knowledge. Similarly, Ruiz de Arcaute et al. [98] also successfully demonstrated MIA attacks against decision trees and other tree-based models. In contrast, Fredrikson et al. [99] address privacy issues of ML-as-a-Service platforms like BigML. More precisely, they introduced new model inversion attacks and demonstrated that these can be used to infer sensitive data features from deployed decision tree models in both a white-box and black-box setting. Furthermore, Mehnaz et al. [100] propose two new model inversion attribute inference attacks in which the adversary's goal is to learn some sensitive attribute about an individual whose data is within the training set. Once again, the attacks could be successfully applied against decision trees. All presented research works provide evidence that decision tree models are highly vulnerable to inference attacks, highlighting serious privacy issues and stressing the relevance of appropriate protection mechanisms.

3.1.4 RANDOM FORESTS

3.1.4.1 Accountability and Explainability

Random forest is an ML method that uses an ensemble of decision trees to combine individual tree predictions into a strong aggregated prediction that relies on the wisdom of the crowd [86] (see Section 2.2.4). Hence, as an ensemble method, a random forest aggregates a large number of weak trees into a strong predictive model that performs better than individual trees but loses the intrinsic interpretability of the latter. Although each decision tree represented in a random forest is inherently interpretable (see Section 3.1.3), the aggregated predictions of the ensembled random forests model are neither interpretable nor explainable, since a random forest as an ensemble method is not differentiable. Furthermore, due to the highly parallel character of a random forest, individual decisions are no longer comprehensible for human users due to the sheer complexity, which causes random forests to be perceived as opaque black-box models. In conclusion, a random forest does not provide local explainability of

⁷ BigML: <https://bigml.com/>, as of date 15.05.2022



D2.1: Accountability and Resilience Analysis

individual predictions. To counteract this problem, post-hoc XAI methods are typically applied to provide explanations and accountability for individual predictions of a random forest. As shown in Table 1, local model-agnostic XAI methods like LIME, SHAP, or counterfactual explanations can be used in this context.

Although individual predictions of a random forest are no longer transparent or comprehensible, Breiman [19] has already shown in his visionary paper that the individual underlying decision trees of a random forest can be used to represent the importance of input features accurately. Thus, a trained random forest can provide indications of the most relevant input features in a dataset with respect to the output variable. This allows users to develop an intuition for the dataset. Hence, a random forest provides a simple form of global explainability. As can be observed from Table 1, other global XAI methods such as permutation feature importance or PDP are also applicable to random forests to explain a trained model globally. However, it must be mentioned that these methods show strong similarities to the idea of the variable importance described by Breiman.

3.1.4.2 Resilience

Similar to decision trees, many recent studies showed that also random forests as tree-based ensemble methods are vulnerable to adversarial attacks, which poses serious security concerns in their application and calls for appropriate countermeasures. For example, Takiddin et al. [101] demonstrated successful poisoning attacks against random forest models in the context of electricity theft detection. Specifically, they were able to show a significant reduction in the accuracy of the RF-based theft detector by injecting malicious data into the dataset. To counteract this vulnerability, the authors recommend using ensemble averaging in order to build more robust detectors by averaging the outputs of different detectors. In [106], Dunn et al. also demonstrated successful poisoning attacks against random forest models in the context of IoT systems. Their findings suggest that the random forest model's accuracy drop is proportional to the rate of poisoned data. As a final example of successful poisoning attacks against random forest models, we would like to mention the paper [91] already discussed in Section 3.1.3.2. Besides data injection attacks against decision trees, Alhajjar et al. [91] also demonstrated the applicability of this attack to random forest models. However, the authors also mention that attacks against ensemble methods are, on average, less effective meaning ensemble methods are more resilient to adversarial attacks than single decision trees.

In addition, the vulnerability of random forest models against evasion attacks has been demonstrated many times. For example, two algorithms were proposed by Kantchelian et al. [94], one that finds an optimal evading instance based on a mixed-integer linear program and one that is computationally faster but does not find an optimal instance. The authors applied these attacks against random forest models and their results show that minimal input perturbations, especially regarding L_1 - and L_2 -norm, suffice in order to obtain a different classification result, demonstrating the vulnerability against evasion attacks. As another example, we want to refer again to the already discussed work [91]. Besides the other attack variants already mentioned above, the paper from Newaz et al. [89] also studies evasion attacks against smart healthcare systems based on random forests. They demonstrated that random forest models are vulnerable to Zeroth Order Optimization attacks, as they were able to craft adversarial examples successfully. Finally, Alhajjar et al. [91] could also show in the already presented work that random forests are vulnerable to various adversarial generation methods



D2.1: Accountability and Resilience Analysis

in the context of NIDSs, providing evidence for the vulnerability of random forests against evasion attacks.

Recent studies have also shown the vulnerability of random forests against data inference attacks. For example, Ruiz de Arcaute et al. [98] successfully applied membership inference attacks against random forest models, suggesting their vulnerability to this kind of attack. As another example, in [102], Luo et al. successfully applied a feature inference attack against a random forest model in a federated learning context, where the performed attack was based on so-called generative regression networks. However, in order to confirm these early findings, further research effort is desirable.

In contrast, model stealing attacks against random forest models seem to be a field of research that has not received much attention so far. In fact, we could only identify one paper from Liu et al. that successfully applied model stealing attacks to random forests [102]. This is probably due to the fact that random forests as an ensemble method provide only non-differentiable and non-decomposable aggregated predictions of individual decision trees, which only allows an imprecise estimation of the learned target function [96]. This may be a reason that renders random forest models relatively resilient to this type of attack. More intensive research is needed to confirm this assumption and is, therefore, a recommendation for future work.

3.1.5 GRADIENT-BOOSTED TREES AND XGBOOST

3.1.5.1 Accountability and Explainability

Gradient-boosted tree and XGBoost models do not provide local explainability by default. These models consist of an ensemble of many decision trees, in which each decision tree is an inherent interpretable model (see Section 3.1.3). While the prediction from each decision tree used in a gradient-boosted tree model is explainable, the overall prediction provided by the ensemble of trees is not locally explainable. The reason is that each tree has its own decision path that is independent from the path of other trees, and it is not possible to identify the exact contribution of each feature in a given prediction.

On the other hand, gradient-boosted tree and XGBoost models are globally explainable, meaning that the importance of each feature in the model and for predictions in general can be known. Based on the ensemble of trees learned during training, it is possible to determine how important each feature is in the model that has been learned. In fact, the global explainability of gradient boosted trees is sometimes used for determining feature importance and performing feature selection.

The prediction of gradient-boosted tree models can be nevertheless locally explained using additional post-hoc explainability approaches. One commonly used approach for gradient-boosted trees is the Breakdown method. It measures how much a given prediction changes when the value of a single feature changes. It computes the difference between smooth prediction scores from the model while modifying each feature individually and it infers the corresponding contribution of each feature in the final prediction. This way, it is possible to know the exact weight of each feature on a specific prediction, providing this local explainability. In particular, Table 1 shows the specific XAI methods that can be used to provide



D2.1: Accountability and Resilience Analysis

local explainability for Gradient-boosted Trees and XGBoost are LIME, SHAP, or counterfactual explanations.

3.1.5.2 Resilience

Gradient-boosted tree and XGBoost models provide good overall resilience to adversarial ML attacks. Although some authors [105] [91] demonstrated that ensemble classifiers can be vulnerable to poisoning attacks that degrade the overall accuracy of the model, these attacks are rather ineffective against gradient-boosted tree models because these models do not generalize over errors in the training data. On the other hand, their tendency to overfit the training data and their low generalization ability makes them vulnerable to backdoor attacks, where only a few targeted errors are aimed for during inference.

Two main characteristics of gradient-boosted tree and XGBoost models makes them relatively resilient to evasion attacks. The first is their composition of many simple models that have different contributions in each prediction. As we discussed, each model is independent and many of these models must be fooled for the ensemble to provide an incorrect prediction. It is difficult to craft an adversarial example that would consistently fool many independent models [104]. Evasion attacks against these models are even complex in a black box setting where an attacker would not know how many weak models compose the models and must be fooled. The second characteristic is that gradient-boosted tree models are non-differentiable, i.e. it is not possible to compute a gradient for the loss of these models that can be propagated back to their inputs. The most effective evasion attacks to craft adversarial examples rely on gradient loss computation and backpropagation of this gradient to algorithmically modify adversarial examples. None of these effective attacks can be applied to gradient-boosted tree models because they are non-differentiable. Black box evasion attacks and white box evasion attacks on a surrogate differentiable model can still be performed but their effectiveness is lower than that of white box evasion attacks. Despite the discussed relatively high resilience of gradient-boosted trees against evasion attacks, some studies have practically demonstrated that these still can be used against gradient-boosted trees and ensemble classifiers [94] [91] [95]. As a countermeasure, Kantchelian et al. [94] suggest Adversarial Boosting to create robust gradient-boosted trees that are more resilient to evasion. Similarly, Cheng et al. [92] propose a method to train XGBoost models robust against evasion attacks.

On the other hand, early research results have shown that gradient-boosted tree models are vulnerable to model stealing attacks, some work being able to steal the functionality of such models using moderate number of queries [103]. Although the vulnerability of these models to data inference attacks has not been well studied yet, the authors of [107] already found that gradient boosted models are less vulnerable to MIA attacks. Nevertheless, we conclude that the vulnerability of gradient-boosted trees and XGBoost to model stealing and data inference attacks is a property that needs to be explored further in future work.

3.1.6 BAYESIAN NETWORKS

3.1.6.1 Accountability and Explainability

The main objective of using state-of-the-art XAI methods is to provide informative answers of “why”, “what-if” or “how” questions to both AI developers and end users. Bayesian networks,



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

which are probabilistic graphical models built on expert knowledge and statistical data, can manage uncertainty, and provide global explanation in the model, reasoning and evidence. Reasoning in Bayesian networks is therefore often referred to as “what-if” questions of an ad-hoc scenario. The flexibility of a Bayesian network allows for these questions to be predictive, diagnostic, and inter-causal [108]. According to [108], explanations in Bayesian Networks can be classified into 3 categories: explanation of the evidence, explanation of the model, and explanation of the reasoning. Apart from well-established reasoning methods, the probabilistic framework of a Bayesian network also allows for global explanations in evidence.

Although BNs are suitable to encapsulate a complex decision-making process thanks to the graphic models representing probabilistic priors and the interdependencies between variables, the local explainability of BNs is still limited. For instance, to explain a prediction of a disease based on some observed evidence, we need to consider not only the most significant evidence, but also understand how this evidence affects the probability of this disease through unobserved variables. Furthermore, BNs lack interpretation and explanation due to complex and indirect relationships between nodes, especially in the big DAG graph. Therefore, to increase the level of trust, the local explanation of inner workings of BNs must be taken into account.

In general, the use of Bayesian networks for explainability is still ad-hoc and not as well organised as XAI methods in the literature. Specifically, not only are the inner workings of Bayesian networks seeming complicated to most end-users, but the explanation of probabilistic reasoning is also challenging and as such results appear to be counter-intuitive or wrong. In the literature, existing work improved the explainability of Bayesian networks mainly based on graphical, visual aids or natural language approach [109]. Thus, the explainability of Bayesian networks still needs to be improved.

3.1.6.2 Resilience

Similar to other AI algorithms, such as Support Vector Machines or Neural Networks, Bayesian Networks are vulnerable against data poisoning attacks that aim at corrupting the AI model by contaminating the data in the training phase. In [110], the authors studied the robustness of BNs structure learning algorithms against traditional one-step data poisoning attacks by investigating two subclasses of data poisoning attacks-model invalidation attacks and targeted change attacks. In model invalidation attacks the model is invalid due to the poisoning of its training dataset. In comparison, by targeted change attacks the goal of the attacker is to create or remove a link in the BNs graphical model. In [111], the same authors further investigated long-duration data poisoning attacks against Bayesian network structure learning algorithms.

Unfortunately, to the best of our knowledge, no relevant literature could be identified that has studied the vulnerability of Bayesian network models to evasion attacks, model stealing attacks, and data inference attacks. Since this seems to be a not well-studied research field, we cannot conclude evidence-based statements about the resilience of Bayesian networks to this kind of attacks. Therefore, we recommend profound research efforts in this direction for future work to obtain better knowledge about the resilience of Bayesian networks.



3.2 ACCOUNTABILITY AND RESILIENCE REQUIREMENTS FOR THE SPATIAL USE CASES

After having analysed the identified ML algorithms regarding their accountability and resilience, we will review the four ML-based SPATIAL use cases for which the application of the analysed algorithms is of significant importance. In this context, we aim to stress the relevance of accountable and resilient decision-making models to these practical use cases. However, we would like to mention that we only provide a short summary of the use cases here. We refer the interested reader to SPATIAL deliverable D1.1 and future technical deliverables for more details.

3.2.1 USE CASE 1: PRIVACY-PRESERVING AI ON THE EDGE AND BEYOND

Our first use case - provided by Telefonica Investigacion Y Desarrollo SA (TID) - is an edge-based federated learning platform which involves numerous client devices and trains an ML model in a distributed and privacy-preserving manner. The use case envisions an environment where multiple machine learning applications are using personal data at a large scale. While such large-scale machine learning applications have been predominantly centralized (i.e. all data of users/clients/devices need to be uploaded to cloud platforms for learning), the recent advances in Federated Learning (FL) allows to build ML models in a decentralized fashion close to users' data, without the need to collect and process them centrally.

We aim to design the platform to be able to train all the ML models described above through Federated Aggregation, as a solution to privacy of user data. As Federated Aggregation is the key process of the use case, we provide a general description of it as the requirement.

Any ML algorithm whose objective function can be written as a finite-sum objective of the following form can be employed in our platform:

$$\min_{w \in \mathbb{R}^d} f(w),$$

in which $f(w)$ is defined as:

$$f(w) = \frac{1}{n} \sum_{i=1}^n f_i(w)$$

Here, n denotes the number of samples across all clients and $f_i(w)$ is defined as loss of prediction on seen examples, i.e., $l(x_i; y_i; w)$, using the trained model with parameters w . All data available in the system (n data points) are partitioned over K clients, each client k with a subset of indices P_k . Then, the problem objective can be rewritten as:

$$f(w) = \sum_{k=1}^K \frac{n_k}{n} F_k(w), \text{ where } F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w)$$

Each client k executes, for example, a stochastic gradient decent (SGD) step (iteration t) on the local data available, $g_k = \nabla F_k(w_t)$. Assuming $C=1$, at iteration t , the server aggregates all gradients and applies the update on the global model: $w_{t+1} \leftarrow w_t - \eta \sum_{k=1}^K \frac{n_k}{n} g_k$,



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

since $\sum_{k=1}^K \frac{n_k}{n} g_k = \nabla f(w_t)$ where η is a fixed learning rate. Equivalently, every client can perform the update as: $w_{t+1}^k \leftarrow w_t - \eta g_k$ and the global model is $w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$.

In fact, this process can be repeated for $t \in E$ iterations locally per client, before sharing models with the server in R rounds. Therefore, the client can iterate the local update $w^k \leftarrow w^k - \eta \nabla F_k(w^k)$ for E times, before the aggregation and averaging at the central server, per round r : $w_{r+1} \leftarrow w_r - \eta \sum_{k=1}^K \frac{n_k}{n} g_k$.

It becomes apparent that factors such as E iterations per client, C clients participating in each round, and R rounds executed can have high impact on model performance, and communication cost incurred in the infrastructure to reach it. These factors have to be carefully chosen based on an examination of the impact.

3.2.2 USE CASE 2: IMPROVING EXPLAINABILITY, RESILIENCE AND PERFORMANCE OF CYBERSECURITY ANALYSIS OF 4G/5G/IoT NETWORKS

We are living in Industry 4.0 and the Internet of Things is undoubtedly a critical component of the fourth industrial revolution. While there are so many factors contributing to this rise, one of the most important is the development of 4G/5G technologies, whose fast speed allows IoT devices to produce and transmit data faster than ever. The combination of 4G/5G, IoT, and AI brings us a complete solution to deal with such a flood of data, as the digital information collected by connected devices and 4G/5G smartphones can now be efficiently contextualised and analysed by ML technologies for automated decision making. However, existing AI-based systems have three major issues in the 4G/5G/IoT context: (1) lack of real-world datasets; (2) lack of explainability; and (3) lack of resilience against adversarial attacks. We therefore provide this use case to tackle the above challenges by considering security, explainability and resilience requirements during the design and development of our AI-based components and then evaluating our AI models on a real testbed regarding those requirements. Concretely, our main objectives are (1) producing real-world datasets, especially for 4G/5G and encrypted network traffic, as training datasets to improve accuracy of AI models; (2) enabling explainability features of existing AI algorithms in our different AI-based systems, such as Montimage Monitoring Tool (MMT)-Probe for anomaly detection and MMT-RCA for Root Cause Analysis (RCA); and, (3) considering the security threats, such as model evasion, model poisoning or backdooring, that emerge from the rapid adoption of AI algorithms in 4G/5G/IoT networks.

The growing popularity of traffic encryption increases user security and privacy at the individual level, but also becomes a big challenge for performing traffic analysis. This raises the need for advanced analysis techniques based on other criteria, such as network packet and flow behaviour analysis. With the introduction of network encryption protocols, such as Transport Layer Security (TLS), the accuracy and efficiency of conventional Network Intrusion Detection Systems (NIDS) using rule and signature-based monitoring detection methods is greatly reduced without being able to analyse packets' payload. Moreover, the variety and dynamicity of network malware poses a significant challenge on traffic monitoring tools in terms of flexibility and generalisation of their algorithms. One of the most emerging solutions for these problems is applying Machine Learning for the analysis.



SPATIAL project is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101021808.

D2.1: Accountability and Resilience Analysis

In the 5G/IoT context, an early detection and prediction of potential anomalous behaviours in the network enables fast reaction to them, preventing financial loss, malicious damage and service degradation. AI has been shown to help detect hidden or abnormal traffic patterns that can lead to security threats or service unavailability in 5G/IoT networks. For instance, [54] leverages the clustering algorithm DBSCAN to effectively detect anomalies caused by radio attenuation and Software Defined Network (SDN) misconfiguration for self-healing of 5G Radio Access Networks (RAN). [55] applies different ML models to predict attacks and anomalies in IoT systems and finds that Random Forest techniques perform comparatively better than others.

RCA plays a vital role in the Network-/System-Management process to accurately identify the cause of faults or security incidents in different domains, such as IT operations and telecommunications. The root cause diagnosis becomes highly intractable or even impossible because of the complexity and heterogeneity of emerging mobile networks (e.g. introducing virtualized functions, Software-Defined Networking), coupled with the increasing number of Key Performance Indicators (KPIs) and data related to end-users, devices, services and networks. Also, human-based mitigation actions become more challenging and time consuming in complex 5G/IoT systems. This leads to the need of an automated tool helping humans to troubleshoot a system and determine which events are causally connected and which are not. AI has been recognized as an appealing option for fostering automated RCA, thanks to its ability to process a large amount of data, uncover complex non-linear relationships within the data, and deliver faster and accurate conclusions.

Our real pilot corresponds to a 4G/5G/IoT solution consisting of an eNodeB/gNodeB based on a Software-Defined Radio, a portable 5G Core solution, and the MMT security monitoring framework. All applications discussed above involve different AI techniques for different security purposes, thus explainability, resilience and accountability are three critical and important aspects in the Use Case 2. Specifically, the cybersecurity analysis and protection of the network, encrypted traffic analysis and RCA developed in our security monitoring framework will employ advanced AI techniques and algorithms to validate the improvements, especially in terms of accountability, resilience and transparency. Our current feature selection, similarity learning and Bayesian networks techniques may need to be improved to make them more resilient to adversarial attacks, transparent and explainable, and privacy-aware. Furthermore, we will also consider the trade-off between the accuracy of AI models and other aspects in the context of the Use Case 2.

3.2.3 USE CASE 3: ACCOUNTABLE AI IN EMERGENCY eCALL SYSTEM

As part of Use Case 3 of SPATIAL project, the Fraunhofer Institute for Open Communication Systems (FOKUS) will investigate the design and implementation of an automated, effective, accountable, and privacy-preserving AI-based system that operates on heterogeneous eHealth data and can accurately recognize emergency situations. The described AI functionalities will be integrated into a modern, IP-based, rich-media emergency communication system, which will enable automated, well-informed, and explainable automated emergency calls (eCalls). Precisely, FOKUS will build on the results of the EU-funded H2020 project EMYNOS⁸ (nExt

⁸ EMYNOS: <https://www.emynos.eu/>, as of date 26.04.2022



D2.1: Accountability and Resilience Analysis

generation eMergencY commuNicatiOnS), which proposed a specification for a Next Generation 112 (NG112) emergency communication system. NG112 represents an evolution of traditional phone-based emergency communication systems. It aims to overcome the limitations of legacy systems such as the transmission of caller location, the forwarding of emergency calls, or the integration of sensor data and IoT architectures.

In this context, EMYNOS designed, defined, and implemented a platform for VoIP-based⁹ bidirectional emergency communication that enables to share additional information, e.g. the location or sensor data of a patient, directly with the emergency call centre during an emergency call. Based on this system, Use Case 3 aims to fully automate the process of emergency detection and the ensuing triggering of an emergency call by collecting, monitoring, and analysing diverse and relevant health data such as a patient's blood pressure, heart rate, oxygenation, blood glucose levels, or body temperature. Effective AI methods and Machine Learning algorithms will be applied to analyse this heterogeneous sensor data. More specifically, high-performing supervised Machine Learning algorithms, such as Random Forests, Deep Neural Networks, or Recurrent Neural Networks, as well as unsupervised algorithms, such as Autoencoders, could be applied to detect anomalies in the gathered eHealth sensor data.

Since emergency communications systems and their applications are to be considered safety-critical infrastructures, the accountability and resilience play a decisive role in the design and integration of such systems and applications. In this context, particularly resilience is of utmost importance. In order to protect the security infrastructure and its applications and to guarantee continuous functioning of the underlying mechanisms, it must be ensured that the developed systems and applications are adequately protected against cyberattacks and can withstand them. Regarding the AI-based system to be developed in Use Case 3, special focus must be placed on resilience against adversarial ML attacks. On the one hand, the system must be resilient against data poisoning and backdoor attacks. This ensures an uncompromised training process of the underlying ML model and is a prerequisite to guarantee that the ML-based system effectively and reliably recognizes emergency situations. On the other hand, the developed system must also be protected against evasion attacks, which aim to compromise the ML-based system at interference time. For example, adversarial examples could be used to trigger hoax calls in Use Case 3, which overload the emergency communication system and block it for serious, urgent emergencies. Besides the attack vectors mentioned above, it must also be taken into account that the eHealth data collected and processed in Use Case 3 must be considered as highly sensitive personal data and thus requires strong protection. Therefore, the ML-based system must also be resilient to and provide countermeasures against data interference and model stealing attacks. These attacks attempt to obtain information about the training data and used model parameters, which could lead to privacy leaks and intellectual property violations. Besides countermeasures against these attacks, additional privacy-preserving methods, such as differential privacy, should be considered due to the highly sensitive nature of the processed data in this use case.

In addition to resilience, accountability and explainability are essential aspects in the context of Use Case 3 and the automated AI-based emergency call functionality. If the system to be developed not just detects emergency situations but also provides explanations of the decision-

⁹ Here, VoIP denotes Voice over IP, in which IP stands for Internet Protocol.



D2.1: Accountability and Resilience Analysis

making process by comprehensibly presenting why a situation is to be classified as an emergency, not only patients but also emergency call centre personnel and doctors can benefit. On the one hand, the transmission of sensor data and XAI explanations enables emergency call centre personnel to assess the situation better, initiate more targeted and effective medical measures, and recognize and reject hoax calls. On the other hand, thanks to the more transparent information available, doctors can better prepare themselves for the emergency situation and, if necessary, already instruct first responders. Patients also benefit from these improvements, as they may receive faster and more targeted medical assistance. Thus, all involved actors benefit from an increased explainability of the system to be developed in Use Case 3.

3.2.4 USE CASE 4: RESILIENT CYBERSECURITY ANALYTICS

The resilient cybersecurity analytics use case studies two representative cybersecurity applications that use ML for automated decision making. The first application relates to the automated detection of malicious documents using a supervised ML classifier. The second application relates to modelling different system behaviours, also using a supervised ML classifier, to identify the type of unknown hosts. In both these use cases, the ML classifier prediction triggers an automated response without human supervision, i.e. block the access and prevent from opening a document detected as malicious and respectively automatically applying a set of security rules based on the identified host type. These automated responses have consequences on the protected system and its users. They are responsible for preventing attacks from compromising the protected system and they can change the user experience, potentially interfering with how the user wants to operate the system. Consequently, the impact from incorrect decisions made by the ML classifier can be critical. The accountability of the ML classifiers and the explainability of their decision are important to justify (and cope with), the potential compromise of the protected system or the inability to use it as intended. The explanation for incorrect predictions can be used to improve the performance of ML-based detection systems and to fix their weaknesses.

Beyond our two applications, explaining the prediction of ML models used for cybersecurity analytics is also highly relevant when these models support human decision. Explanation of prediction provides understanding to human operators. For instance, managed Detection & Response services require an understanding of the detected attacks to respond to it in the most appropriate manner: to block the attack and to recover from it. When new attacks are detected, it is also useful to create an intelligence about them and to infer the vulnerabilities they exploit. It prevents them from happening again in the future. The same requirement applies to forensic analysis where the main goal is to find the root causes of an attack. Explainable ML models support this investigation and build an understanding of security threats and of the vulnerability they exploit. It also enables to learn more information about identified attacks, classify them in categories and provide global trends about the type of attacks happening at a certain point in time.

Resilience is naturally a primary requirement for resilient cybersecurity analytics, more important than explainability. In this use case, resilience relates mostly to the ability to resist adversarial machine learning attacks that can be launched by attackers wanting to compromise the system protected by a resilient cybersecurity analytic solution. The ML models used for resilient cybersecurity analytics have the main goal to counter cyberattacks, meaning that they



D2.1: Accountability and Resilience Analysis

exist only because attackers exist who want to compromise the protected system. As such, these ML models are natural targets for these attackers, wanting to circumvent and fool them. The accuracy and the availability of ML models' predictions must remain consistently high, regardless of any attackers' action. ML models must be at least as secure and resilient to adversarial ML attacks as the system they protect, i.e., they must not represent a weak link that can be easily compromised by an attacker. The security of ML models used in cybersecurity applications is additionally critical for the image of the cybersecurity vendor deploying them. If and ML-based security function gets compromised by an attacker, it is detrimental to the image of the vendor supposed to enforce the security of the system.

The resilience of ML models to the most common adversarial ML attacks, like model evasion, model poisoning or backdooring [56] must be enforced to meet the resilience requirement. Resilience to attacks must be considered during the design of ML models and it must be completed with defences specifically designed to protect from these attacks. Resilience guarantees the integrity and the availability of the model, which are paramount properties to ensure the cybersecurity analytic functions.

The resilience requirement is not limited to the only protection of the ML model properties. It is extended to protecting the confidentiality of the data that ML models used for resilient cybersecurity analytics are trained with. This data can be privacy-sensitive, depict user's habits or contain personally identifiable information. For instance, internet browsing history is used to detect malicious websites, email content is used to detect spam and phishing emails, information about the file system and process launch is used to detect malicious programs, etc. These types of data contain private information from users of the protected systems, and they must remain confidential. ML models can leak information about the data they use during training through privacy attacks [57] such as model inversion or membership inference attacks. Resilience must enforce that data leakage from ML models is not possible and no privacy-sensitive information should be possible to infer due to access to the ML model or its output predictions by an attacker.

Considering the strong requirements for resilience and high accuracy, resilient cybersecurity analytics often use ensemble models such as Random Forest or Gradient-boosted trees. These types of models are recognized for their ability to reach top accuracy together with a relatively high resistance against adversarial machine learning attacks. On the downside, ensemble models are usually not locally explainable, but this use case can accommodate to prioritize accuracy and resilience over explainability. Our two applications in this use case will implement ensemble classifiers such as Random Forest and XGBoost.



4 DISCUSSION AND RECOMMENDATIONS

In this section, we discuss the main findings of our conducted analysis regarding the accountability and resilience of the discussed ML models. In this context, we also want to identify recommendations and guidelines that can help to improve the accountability and resilience of the examined models.

4.1 FINDINGS ON ACCOUNTABILITY

The analysis performed in this document reveals that many of the high-performing ML models exhibit restricted local and global explainability. This shortcoming limits their accountability in security-critical and safety-critical applications. This is what we expected since many of the discussed ML algorithms (i.e., DNNs, SVMs, random forests, gradient-boosted trees, and XGBoost) are perceived as opaque black-boxes due to their enormous complexity. Therefore, their functionality is no longer transparent and comprehensible to humans. However, since understanding the functionality of a system is a basic requirement to make it accountable for its behaviour, an apparent conflict regarding limited accountability arises. Furthermore, the conducted analysis shows that intrinsic interpretable ML models like decision trees exist, which still offer a high degree of accountability. Decision trees offer an intrinsic trade-off between interpretability and expressive power and are, therefore, they are a good choice for applications that require a high degree of accountability. The downside of using decision tree is that they show a significantly reduced performance compared to the discussed black-box models. However, since efficiency and performance are of crucial importance for many applications, the use of black-box models will probably remain preferred in the development of ML-based applications.

XAI methods can be used to provide explainability and thus improve accountability

To enable accountability of systems based on black-box ML models, state-of-the-art XAI methods can be applied. In this context, methods exist that aim to explain individual decisions of black-box ML models and thus make individual decisions comprehensible and accountable. Such methods are referred to as local XAI methods and include, among others, methods like LIME, SHAP, or LRP (see Table 1). In contrast, global XAI methods like partial dependence plots or permutation feature importance exist. As the name implies, these try to provide a global understanding of the functioning of machine learning models. Typically, global XAI methods aim to explain the global relationship between the relevance of specific input features and the model prediction. In this context, the analysis also revealed that some black-box models exhibit intrinsic global explainability. For example, random forest models allow for an estimation of the relevance of input features to the predictions through feature importance [19].

The best suited XAI method not only depends on the ML model but also on the application

Table 1 summarizes the discussion about applicable XAI methods and presents an overview of the ML models analysed in this deliverable and their suitable XAI methods. However, it should be noted that while many of the XAI methods presented are theoretically applicable to certain ML algorithms, their practical applicability depends on the exact problem at hand. For example, local explainability methods such as LRP or occlusion sensitivity are typically employed for



D2.1: Accountability and Resilience Analysis

explaining DNN models that process image data (e.g., CNNs). If, on the other hand, a DNN is used to perform network intrusion detection based on network traffic, feature relevance methods such as PDP or counterfactual explanations may be better alternatives.

Expected explanations are subjective to the problem at hand and depends on user knowledge

These two examples illustrate another challenge in applying XAI methods: explanations are highly subjective to the problem at hand and the knowledge of the targeted user. For example, non-expert users typically expect easy-to-understand low-level explanations such as the visual explanations provided by LIME or LRP. In addition, counterfactual explanations can be a good way to explain ML predictions to non-experts. Furthermore, domain-level experts may require more detailed explanations. Typically, they expect explanations that represent more complex relationships. Here, depending on the application, global XAI methods like permutation feature importance may be more relevant.

Developers need to find a balanced trade-off between performance and accountability

To conclude the discussion about the accountability of ML models, it can be summarized that the selection of a suitable machine learning algorithm for a specific application always depends on many factors. Typically, the correct choice implies a trade-off between accountability and performance, for which a fair balance has to be found. In general, it can be observed that the more performant an ML model is, the less explainable and understandable it is, causing accountability to suffer. This trade-off has to be taken into account by developers of ML-based applications. The right ML algorithms have to be chosen carefully in order to find the best possible trade-off between the performance and accountability of an application.

4.2 FINDINGS ON RESILIENCE

In addition to accountability, this deliverable also analysed the resilience of the presented ML algorithms against adversarial attacks. We found from the existing scientific literature (as summarised in Table 2) that all of the discussed ML algorithms are vulnerable to adversarial attacks such as poisoning attacks, evasion attacks, data inference attacks, and model stealing attacks. Only for gradient-boosted trees and Bayesian networks, no research studies could be identified for some of the investigated attack variants. More precisely, no literature could be identified that investigates their resilience against data interference attacks for gradient-boosted trees. For Bayesian networks, the available literature seems to be even more limited. No literature could be found for Bayesian networks regarding their vulnerability to evasion attacks, data inference attacks, and model stealing attacks. However, we do not conclude that these algorithms are resilient against these attacks. Instead, we suspect that this is still a not well-studied area of research, and further investigations are needed. In fact, it can be concluded from our analysis that all discussed ML models are vulnerable to certain adversarial attacks. This can lead to new attack vectors for ML-based systems, which raises new security concerns in integrating ML algorithms into traditional systems.

Attack success rate depends on model visibility, attacker's knowledge, and dataset

As a further finding of the performed resilience analysis, we want to mention that the attacker's success rate depends strongly on the visibility of the attacked ML model. In general, white-box



D2.1: Accountability and Resilience Analysis

models are more prone to adversarial attacks than black-box models. In the former case, the attacker has access to the model internals, whereas the attacker has only access to the model output in the latter case. Furthermore, it was found that the chance of success also depends on the attacker's knowledge. For example, in [97], a membership inference attack in a black-box setting was successfully applied to decision trees. The authors showed that the effectiveness of the attack in terms of MIA accuracy strongly depends on the attacker's knowledge but also on the training dataset used. This highlights the importance of protecting access to model internals for a deployed ML in the operational phase.

The effectiveness of adversarial attacks against different ML models is difficult to compare

We would also like to briefly discuss the possibilities of comparing the resilience of different ML models against the attacks. Many of the studies identified in the conducted analysis compare the effectiveness of applied attacks between different ML models. Thereby, the authors sometimes come to contradictory conclusions. For example, the authors of [94] argue that a random forest model is more vulnerable to evasion attacks than other ML models because of the smaller perturbation needed to change the classification outcome. In contrast, the authors of [91] conclude that models based on random forests are more robust as the evasion rate is lower in most experiments. These different conclusions highlight that the efficiency of the attacks against different ML methods cannot generally be compared and evaluated. This is due to the fact that the concrete results and attack success strongly depend on the different application domains, used datasets, and different metrics used to evaluate the performance decrease. Therefore, the effectiveness of the discussed attacks cannot be compared between the different ML models in an application-independent way.

Initial protection and defence strategies against adversarial attacks exist

The analysis also identified some first recommendations and guidelines that offer initial protection and defence strategies against adversarial attacks. For example, it is crucial to verify the data sources and supplied data used during training, in order to protect against poisoning attacks. Furthermore, several suitable anti-poisoning techniques and defence strategies exist against poisoning attacks, such as outlier detection, Reject On Negative Impact defence [74], or data sanitization defences [112] [113]. Moreover, also strategies to protect against evasion attacks exist. For example, the authors of [94] propose Adversarial Boosting, in which the combination of several ML models in an ensemble fashion makes it possible to develop more robust systems. In addition, some software frameworks exist that implement methods to protect against adversarial attacks and enable robust training (sometimes referred to as adversarial training) of ML models, e.g., the Python libraries Foolbox [114] and Adversarial Robustness Toolbox [115]. In this context, when deciding on an ML model, it can be useful to quantify and compare the robustness of different models by using these packages. Despite the guidelines and defence strategies mentioned at the beginning of this paragraph, our analysis findings emphasize the need for modelling attacker capabilities as well as the development and application of more powerful countermeasures and defence strategies.



5 CONCLUSIONS AND OUTLOOK

This deliverable document analysed six different machine learning algorithms concerning their accountability and resilience characteristics. The findings of the conducted analysis will form the basis for further activities in the SPATIAL project, in which resilient accountability metrics will be proposed and integrated into existing ML algorithms. Hence, this document is intended to capture insights regarding current accountability and resilience characteristics and identify challenges towards establishing accountable and resilient AI. The selection of the algorithms analysed in this document is based on their potential relevance for the four SPATIAL use cases, which reflect the domains IoT, 5G, cybersecurity, and eHealth. Specifically, we analysed DNNs, SVMs, decision trees, random forests, gradient-boosted trees, and XGBoost.

In SPATIAL, we understand the accountability of AI as the representation of the AI models in a way that they can be easily understood. Therefore, explainability plays a crucial role in our accountability analysis, as we understand the explainability of ML algorithms as a means to achieve accountability. Thus, we discussed the intrinsic explainability of ML algorithms based on their underlying algorithmic properties. However, since our findings show that many of the discussed models are non-comprehensible and non-transparent black-boxes, we also analysed the applicability of different XAI methods. These can be used to provide both local and global explanations of a black-box ML model, thereby increasing the understanding of its decision-making process. A summary of the identified applicable XAI methods per algorithm is presented in Table 1. *TABLE 1*

Based on our investigation, we can say that the identified state-of-the-art XAI methods can indeed be used to improve the local and global explainability of black-box models. However, we have to mention that the most appropriate XAI method for an algorithm cannot be determined a-priori and depends on both, the task at hand and the user to whom the explanations are addressed. The latter is motivated by the fact that explanations are subjective to the problem at hand and the user's knowledge. It can be summarized that the right choice of the ML algorithm and appropriate XAI methods depends on the tasks' accountability requirements. Since typically high-performing models exhibit less accountability, selecting the right algorithm always represents a trade-off between required performance and accepted accountability, for which the developer has to find a fair balance. For a more detailed discussion of the summarized findings, we refer to Section 4.1.

Besides, the ML algorithms were investigated with respect to their resilience against adversarial attacks (e.g., poisoning attacks, evasion attacks, model stealing attacks, and data inference attacks). In order to draw conclusions about their resilience against these attacks, we have identified recent scientific literature that has looked into the algorithms' vulnerability to these attacks (see Table 2). Our findings show that all ML algorithms discussed are to some degree vulnerable to the studied adversarial attacks. This can lead to new attack vectors for ML-based systems, which raises new security concerns in integrating ML algorithms into traditional systems. In the conducted analysis, it was identified that adversarial attacks can cause significant degradation of model performance (poisoning attacks), serious operational issues (evasion attacks), privacy issues (data inference attacks), and violations of intellectual properties (model stealing attacks). In addition, there are indications that the actual success rate depends on the visibility of the ML model to the attacker (black-box vs white-box setting) as well as the



D2.1: Accountability and Resilience Analysis

used dataset and the concrete application. This highlights the importance of protecting access to the internals of deployed ML models from third parties and potential attackers. Furthermore, this suggests that application-independent comparability of the vulnerability of ML models is difficult, which implies that no general statements can be made about the degree of vulnerability of the models. As a final finding, we note that initial protection and defence strategies against adversarial attacks already exist. These include anti-poisoning techniques such as outlier detection, Reject On Negative Impact defence [74], or data sanitization defences [112] [113]. But also methods like adversarial boosting [94] or robust/adversarial training can provide protection against adversarial attacks. In this respect, some software frameworks already exist that implement some of the aforementioned defence strategies and support ML practitioners, e.g., the Python libraries Foolbox [120] and Adversarial Robustness Toolbox [115]. Again, we refer to Section 4.2 for a more detailed discussion on the mentioned findings.

In conclusion, the findings obtained in this deliverable indicate that the discussed ML algorithms hold different resilience and accountability characteristics. Furthermore, the findings suggest that selecting a suitable ML algorithm always constitutes a trade-off between performance, accountability, and resilience. The problem of finding an optimal balance for this trade-off clearly demonstrates the need for appropriate measures to compare and assess the accountability, resilience, and accuracy of ML models. These aspects will be the focus of further activities in the SPATIAL project, in which resilient accountability metrics will be proposed and integrated into existing ML algorithms.

Finally, we want to mention that we only studied the resilience of ML models against adversarial attacks in this deliverable. However, to achieve trustworthy AI, a more wholesome approach is required that considers the resilience of all components of an ML-based system. This includes the models analysed in this document, but also traditional components and all processing steps of an ML pipeline. These are also aspects that the SPATIAL project will investigate in further activities.



REFERENCES

- [1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, 'A survey of methods for explaining black box models', *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.
- [2] A. Rosenfeld en A. Richardson, "Explainability in human--agent systems", *Autonomous Agents and Multi-Agent Systems*, vol 33, no 6, blI 673–705, 2019.
- [3] Cellan-Jones, R. (2020). Uber's self-driving operator charged over fatal crash. *BBC News*.
- [4] European-Commision (2019). "Ethics guidelines for trustworthy AI."
- [5] Kacianka, S. and A. Pretschner (2021). Designing Accountable Systems. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*: 424-437.
- [6] Schwartz, O. (2019). In 2016, Microsoft's Racist Chatbot Revealed the Dangers of Online Conversation. *IEEE Spectrum*.
- [7] Wieringa, M. (2020). What to account for when accounting for algorithms. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*: 1-18.
- [8] Eigner, Oliver, et al. "Towards Resilient Artificial Intelligence: Survey and Research Issues." 2021 IEEE International Conference on Cyber Security and Resilience (CSR). IEEE, 2021.
- [9] Tesla's Autopilot faces US investigation after crashes with emergency vehicles: <https://www.theguardian.com/technology/2021/aug/16/teslas-autopilot-us-investigation-crashes-emergency-vehicles>, as of date 30.05.2022
- [10] IBM's Watson supercomputer recommended 'unsafe and incorrect' cancer treatments, internal documents show: <https://www.statnews.com/2018/07/25/ibm-watson-recommended-unsafe-incorrect-treatments/>, as of date 30.05.2022
- [11] David E. Rumelhart; James L. McClelland, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, MIT Press, 1987, pp.318-362.
- [12] Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. *Neural Comput* 1997; 9 (8): 1735–1780. doi: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [13] B. Boser, I. Guyon, and V. Vapnik, A training algorithm for optimal margin classifiers, 1992. doi: 10.1145/130385.130401.
- [14] D. Srivastava and L. Bhambhu, Data classification using support vector machine, *Journal of Theoretical and Applied Information Technology*, vol. 12, pp. 1–7, Feb. 2010.
- [15] V. Jakkula, Tutorial on Support Vector Machine (SVM), 2011. [https://www.semanticscholar.org/paper/Tutorial-on-Support-Vector-Machine-\(SVM\)-Jakkula/7cc83e98367721bfb908a8f703ef5379042c4bd9](https://www.semanticscholar.org/paper/Tutorial-on-Support-Vector-Machine-(SVM)-Jakkula/7cc83e98367721bfb908a8f703ef5379042c4bd9), accessed Jan. 27, 2022
- [16] J. R. Quinlan, Decision trees and decision-making, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 339–346, Mar. 1990, doi: 10.1109/21.52545.
- [17] L. E. Raileanu and K. Stoffel, Theoretical Comparison between the Gini Index and



D2.1: Accountability and Resilience Analysis

- Information Gain Criteria, *Annals of Mathematics and Artificial Intelligence*, vol. 41, no. 1, pp. 77–93, May 2004, doi: 10.1023/B:AMAI.0000018580.96245.c6.
- [18] G. Biau and E. Scornet, A random forest guided tour, *TEST*, vol. 25, no. 2, pp. 197–227, Jun. 2016, doi: 10.1007/s11749-016-0481-7.
- [19] L. Breiman, Random Forests, *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [20] T. K. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, Aug. 1998, doi: 10.1109/34.709601.
- [21] Sutton, C., Sindelar, M., & McCallum, A. (2005). Feature bagging: Preventing weight undertraining in structured discriminative learning. Center for Intelligent Information Retrieval, U. of Massachusetts.
- [22] Murphy K. (1998): A Brief Introduction to Graphical Models and Bayesian Networks.
- [23] Bayesian Networks, Michal Horný (2014), online: <https://www.bu.edu/sph/files/2014/05/bayesian-networks-final.pdf>, as of date 30.05.2022
- [24] M. T. Ribeiro, S. Singh, en 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, bll 1135–1144.
- [25] S. Mane en D. Rao, “Explaining Network Intrusion Detection System Using Explainable AI Framework”, arXiv preprint arXiv:2103. 07110, 2021.
- [26] J. Rathod, C. Joshi, J. Khochare, en F. Kazi, “Interpreting a Black-Box Model used for SCADA Attack detection in Gas Pipelines Control System”, in *2020 IEEE 17th India Council International Conference (INDICON)*, 2020, bll 1–7.
- [27] S. M. Lundberg en S.-I. Lee, “A unified approach to interpreting model predictions”, in *Proceedings of the 31st international conference on neural information processing systems*, 2017, bll 4768–4777.
- [28] M. Sarhan, S. Layeghy, en M. Portmann, “An Explainable Machine Learning-based Network Intrusion Detection System for Enabling Generalisability in Securing IoT Networks”, arXiv preprint arXiv:2104. 07183, 2021.
- [29] Ramaravind Kommiya Mothilal, Amit Sharma, Chenhao Tan: Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. CoRR abs/1905.07697 (2019)
- [30] Sahil Verma, John P. Dickerson, Keegan Hines: Counterfactual Explanations for Machine Learning: A Review. CoRR abs/2010.10596 (2020)
- [31] Molnar, C. (2020). Interpretable machine learning. <https://christophm.github.io/interpretable-ml-book/counterfactual.html>, as of date 30.05.2022
- [32] Sandra Wachter, Brent D. Mittelstadt, Chris Russell: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. CoRR abs/1711.00399 (2017)



D2.1: Accountability and Resilience Analysis

- [33] André Artelt, Barbara Hammer: On the computation of counterfactual explanations - A survey. CoRR abs/1911.07749 (2019)
- [34] Susanne Dandl, Christoph Molnar, Martin Binder, Bernd Bischl: Multi-Objective Counterfactual Explanations. PPSN (1) 2020: 448-469
- [35] Kaggle, Tutorial on Permutation Importance, <https://www.kaggle.com/code/dansbecker/permutation-importance/tutorial>, as of date 31.05.2022
- [36] Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously." <http://arxiv.org/abs/1801.01489> (2018)
- [37] Friedman, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (2001): 1189-1232.
- [38] Scikit-learn, Partial Dependence and Individual Conditional Expectation plots, https://scikit-learn.org/stable/modules/partial_dependence.html, as of date 31.05.2022
- [39] Van der Maaten, L. and Hinton, G., 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).
- [40] Hinton, Geoffrey E., and Sam Roweis. "Stochastic neighbor embedding." *Advances in neural information processing systems* 15 (2002).
- [41] Wattenberg, et al., "How to Use t-SNE Effectively", *Distill*, 2016. <http://doi.org/10.23915/distill.00002>
- [42] Karpathy, A. (2014). t-SNE visualization of CNN codes. Retrieved April 4, 2022 from <https://cs.stanford.edu/people/karpathy/cnnembed/>.
- [43] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, Klaus-Robert Müller: Layer-Wise Relevance Propagation: An Overview. *Explainable AI 2019*: 193-209
- [44] Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K. R. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications*, 10(1), 1-8.
- [45] Montavon, G., Lapuschkin, S., Binder, A., Samek, W. and Müller, K.R., 2017. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern recognition*, 65, pp.211-222.
- [46] Velmurugan, M., Ouyang, C., Moreira, C., & Sindhgatta, R. (2020). Evaluating Explainable Methods for Predictive Process Analytics: A Functionally-Grounded Approach. *arXiv preprint arXiv:2012.04218*.
- [47] Resta, M., Monreale, A., & Bacciu, D. (2021). Occlusion-Based Explanations in Deep Recurrent Models for Biomedical Signals. *Entropy*, 23(8), 1064.
- [48] Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3), 247-278.
- [49] Bolei Zhou, A. K., Agata Lapedriza, Aude Oliva, Antonio Torralba "Learning Deep Features



D2.1: Accountability and Resilience Analysis

- for Discriminative Localization", arXiv:1512.04150 [cs.CV], online: <https://arxiv.org/abs/1512.04150>, doi: <https://doi.org/10.48550/arXiv.1512.04150>
- [50] Li, X.-H., et al. (2020). "A Survey of Data-driven and Knowledge-aware eXplainable AI." IEEE Transactions on Knowledge and Data Engineering: 1-1.
- [51] Min Lin et al. (2014). "Network In Network", arXiv:1312.4400 [cs.NE], doi: <https://doi.org/10.48550/arXiv.1312.4400>
- [52] Ramprasaath R. Selvaraju, A. D., Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, Dhruv Batra (2017). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization."
- [53] Ramprasaath R. Selvaraju, A. D., Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, Dhruv Batra (2017). "Grad-CAM: Why did you say that?"
- [54] J. Ali-Tolppa, S. Kocsis, B. Schultz, L. Bodrog and M. Kajo, "Self-healing and Resilience in Future 5G Cognitive Autonomous Networks." ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K), 2018.
- [55] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches." Internet of Things, Volume 7, 2019
- [56] Papernot, Nicolas, Patrick McDaniel, Arunesh Sinha, and Michael P. Wellman. "Sok: Security and privacy in machine learning." In 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 399-414. IEEE, 2018.
- [57] Yeom S, Giacomelli I, Fredrikson M, Jha S. Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st Computer Security Foundations Symposium (CSF) 2018 Jul 9 (pp. 268-282). IEEE
- [58] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, 'Transferable clean-label poisoning attacks on deep neural nets', in the International Conference on Machine Learning, 2019, pp. 7614-7623
- [59] A. Shafahi et al., 'Poison frogs! targeted clean-label poisoning attacks on neural networks', Advances in neural information processing systems, vol. 31, 2018.
- [60] Y. Liu, X. Ma, J. Bailey, and F. Lu, 'Reflection backdoor: A natural backdoor attack on deep neural networks', in European Conference on Computer Vision, 2020, pp. 182-199.
- [61] K. Kurita, P. Michel, and G. Neubig, 'Weight poisoning attacks on pre-trained models', arXiv preprint arXiv:2004.06660, 2020.
- [62] J. Li, Y. Yang, J. S. Sun, K. Tomsovic, and H. Qi, 'Towards Adversarial-Resilient Deep Neural Networks for False Data Injection Attack Detection in Power Grids', arXiv preprint arXiv:2102.09057, 2021.
- [63] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, 'The limitations of deep learning in adversarial settings', in 2016 IEEE European symposium on security and privacy (EuroS&P), 2016, pp. 372-387.
- [64] N. Carlini and D. Wagner, 'Towards evaluating the robustness of neural networks', in the 2017 IEEE Symposium on Security and Privacy (SP), 2017, pp. 39-57.



D2.1: Accountability and Resilience Analysis

- [65] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, 'Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models', in the Proceedings of the 10th ACM workshop on artificial intelligence and security, 2017, pp. 15–26.
- [66] W. Brendel, J. Rauber, and M. Bethge, 'Decision-based adversarial attacks: Reliable attacks against black-box machine learning models', arXiv preprint arXiv:1712. 04248, 2017.
- [67] X. He, J. Jia, M. Backes, N. Z. Gong, and Y. Zhang, 'Stealing links from graph neural networks', in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2669–2686.
- [68] V. Duddu, D. Samanta, D. V. Rao, and V. E. Balas, 'Stealing neural networks via timing side channels', arXiv preprint arXiv:1812. 11720, 2018.
- [69] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, 'Membership inference attacks against machine learning models', in the 2017 IEEE symposium on security and privacy (SP), 2017, pp. 3–18.
- [70] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, 'The secret revealer: Generative model-inversion attacks against deep neural networks', in the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 253–261.
- [71] D. Gopinath, H. Converse, C. Pasareanu, and A. Taly, 'Property inference for deep neural networks', in the 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 797–809.
- [72] F. Khalid, M. A. Hanif, and M. Shafique, 'Exploiting Vulnerabilities in Deep Neural Networks: Adversarial and Fault-Injection Attacks', arXiv preprint arXiv:2105. 03251, 2021.
- [73] S. Weerasinghe, T. Alpcan, S. M. Erfani and C. Leckie, "Defending Support Vector Machines Against Data Poisoning Attacks," in IEEE Transactions on Information Forensics and Security, vol. 16, pp. 2566-2578, 2021.
- [74] B. Nelson et al., 'Exploiting machine learning to subvert your spam filter', LEET, vol. 8, pp. 16–17, 2008.
- [75] B. Biggio, B. Nelson, and P. Laskov, 'Poisoning attacks against support vector machines', arXiv preprint arXiv:1206. 6389, 2012.
- [76] M. James, M. Mruthula, V. Bhaskaran, S. Asha, and Others, 'Evasion Attacks On Svm Classifier', in 2019 9th International Conference on Advances in Computing and Communication (ICACC), 2019, pp. 125–129.
- [77] H. Chen, J. Su, L. Qiao, and Q. Xin, 'Malware collusion attack against SVM: Issues and countermeasures', Applied Sciences, vol. 8, no. 10, pp. 1718, 2018.
- [78] B. Biggio et al., 'Evasion attacks against machine learning at test time', in Joint European conference on machine learning and knowledge discovery in databases, 2013, pp. 387–402.
- [79] B. Biggio et al., 'Security evaluation of support vector machines in adversarial environments', in Support Vector Machines Applications, Springer, 2014, pp. 105–153



D2.1: Accountability and Resilience Analysis

- [80] R. N. Reith, T. Schneider, and O. Tkachenko, 'Efficiently stealing your machine learning models', in Proceedings of the 18th ACM workshop on privacy in the electronic society, 2019, pp. 198–210.
- [81] D. Lowd and C. Meek, 'Adversarial learning', in Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 2005, pp. 641–647.
- [82] M. R. Clark, P. Swartz, A. Alten, and R. M. Salih, 'Toward Black-box Image Extraction Attacks on RBF SVM Classification Model', in 2020 IEEE/ACM Symposium on Edge Computing (SEC), 2020, pp. 394–399.
- [83] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, 'Socinf: Membership inference attacks on social media health data with machine learning', IEEE Transactions on Computational Social Systems, vol. 6, no. 5, pp. 907–921, 2019.
- [84] K. J. Reza, M. Z. Islam, and V. Estivill-Castro, 'Privacy protection of online social network users, against attribute inference attacks, through the use of a set of exhaustive rules', Neural Computing and Applications, vol. 33, pp. 12397–12427, 2021.
- [85] Y. Alufaisan, M. Kantarcioglu, and Y. Zhou, 'Robust transparency against model inversion attacks', IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 5, pp. 2061–2073, 2020.
- [86] L. Gianfagna and A. Di Cecco, Explainable AI with Python. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-68640-6.
- [87] A. B. Arrieta et al., 'Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI', ArXiv191010045 Cs, Dec. 2019, Accessed: Jan. 31, 2022. [Online]. Available: <http://arxiv.org/abs/1910.10045>
- [88] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha, 'Systematic Poisoning Attacks on and Defenses for Machine Learning in Healthcare', IEEE J. Biomed. Health Inform., vol. 19, no. 6, pp. 1893–1905, Nov. 2015, doi: 10.1109/JBHI.2014.2344095.
- [89] A. I. Newaz, N. I. Haque, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, 'Adversarial Attacks to Machine Learning-Based Smart Healthcare Systems', in GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Dec. 2020, pp. 1–6. doi: 10.1109/GLOBECOM42002.2020.9322472.
- [90] J. Chen, M. I. Jordan, and M. J. Wainwright, 'HopSkipJumpAttack: A Query-Efficient Decision-Based Attack', ArXiv190402144 Cs Math Stat, Apr. 2020, Accessed: Apr. 28, 2022. [Online]. Available: <http://arxiv.org/abs/1904.02144>
- [91] E. Alhajar, P. Maxwell, and N. D. Bastian, 'Adversarial Machine Learning in Network Intrusion Detection Systems', ArXiv200411898 Cs Stat, Apr. 2020, Accessed: Apr. 26, 2022. [Online]. Available: <http://arxiv.org/abs/2004.11898>
- [92] H. Chen, H. Zhang, D. Boning, and C.-J. Hsieh, 'Robust Decision Trees Against Adversarial Examples', ArXiv190210660 Cs Stat, Jun. 2019, Accessed: Apr. 11, 2022. [Online]. Available: <http://arxiv.org/abs/1902.10660>
- [93] N. Papernot, P. McDaniel, and I. Goodfellow, 'Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples', ArXiv160507277 Cs, May 2016, Accessed: Apr. 11, 2022. [Online]. Available: <http://arxiv.org/abs/1605.07277>



D2.1: Accountability and Resilience Analysis

- [94] A. Kantchelian, J. D. Tygar, and A. D. Joseph, 'Evasion and Hardening of Tree Ensemble Classifiers', ArXiv150907892 Cs Stat, May 2016, Accessed: Apr. 11, 2022. [Online]. Available: <http://arxiv.org/abs/1509.07892>
- [95] M. Cheng, T. Le, P.-Y. Chen, J. Yi, H. Zhang, and C.-J. Hsieh, 'Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach', ArXiv180704457 Cs Stat, Jul. 2018, Accessed: Apr. 11, 2022. [Online]. Available: <http://arxiv.org/abs/1807.04457>
- [96] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, 'Stealing Machine Learning Models via Prediction APIs', ArXiv160902943 Cs Stat, Oct. 2016, Accessed: Apr. 26, 2022. [Online]. Available: <http://arxiv.org/abs/1609.02943>
- [97] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, 'Demystifying Membership Inference Attacks in Machine Learning as a Service', IEEE Trans. Serv. Comput., vol. 14, no. 6, pp. 2073–2089, Nov. 2021, doi: 10.1109/TSC.2019.2897554.
- [98] G. M. Ruiz de Arcaute, J. A. Hernández, and P. Reviriego, 'Assessing the Impact of Membership Inference Attacks on Classical Machine Learning Algorithms', in 2022 18th International Conference on the Design of Reliable Communication Networks (DRCN), Mar. 2022, pp. 1–4. doi: 10.1109/DRCN53993.2022.9758025.
- [99] M. Fredrikson, S. Jha, and T. Ristenpart, 'Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures', in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver Colorado USA, Oct. 2015, pp. 1322–1333. doi: 10.1145/2810103.2813677.
- [100] S. Mehnaz, N. Li, and E. Bertino, 'Black-box Model Inversion Attribute Inference Attacks on Classification Models', ArXiv201203404 Cs, Dec. 2020, Accessed: May 04, 2022. [Online]. Available: <http://arxiv.org/abs/2012.03404>
- [101] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, 'Robust Electricity Theft Detection Against Data Poisoning Attacks in Smart Grids', IEEE Trans. Smart Grid, vol. 12, no. 3, pp. 2675–2684, May 2021, doi: 10.1109/TSG.2020.3047864.
- [102] X. Luo, Y. Wu, X. Xiao, and B. C. Ooi, 'Feature Inference Attack on Model Predictions in Vertical Federated Learning', in 2021 IEEE 37th International Conference on Data Engineering (ICDE), Apr. 2021, pp. 181–192. doi: 10.1109/ICDE51399.2021.00023.
- [103] G. Liu, S. Wang, B. Wan, Z. Wang, and C. Wang, 'ML-Stealer: Stealing Prediction Functionality of Machine Learning Models with Mere Black-Box Access', in 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Oct. 2021, pp. 532–539. doi: 10.1109/TrustCom53373.2021.00083.
- [104] Barreno, M., Bartlett, P. L., Chi, F. J., Joseph, A. D., Nelson, B., Rubinstein, B. I., ... & Tygar, J. D. (2008, October). Open problems in the security of learning. In Proceedings of the 1st ACM workshop on Workshop on AI Sec (pp. 19-26).
- [105] Biggio, B., Corona, I., Fumera, G., Giacinto, G., & Roli, F. (2011, June). Bagging classifiers for fighting poisoning attacks in adversarial classification tasks. In International workshop on multiple classifier systems (pp. 350-359). Springer, Berlin, Heidelberg
- [106] C. Dunn, N. Moustafa, and B. Turnbull, "Robustness Evaluations of Sustainable Machine Learning Models against Data Poisoning Attacks in the Internet of Things," Sustainability,



vol. 12, no. 16, Art. no. 16, Jan. 2020, doi: 10.3390/su12166434

- [107] G. M. Ruiz de Arcaute, J. A. Hernández, and P. Reviriego, "Assessing the Impact of Membership Inference Attacks on Classical Machine Learning Algorithms," in 2022 18th International Conference on the Design of Reliable Communication Networks (DRCN), Mar. 2022, pp. 1–4. doi: 10.1109/DRCN53993.2022.9758025.
- [108] Derks, Iena Petronella, and Alta de Waal. "A taxonomy of explainable Bayesian networks." Southern African Conference for Artificial Intelligence Research. Springer, Cham, 2021.
- [109] Keppens, Jeroen. "Explainable Bayesian network query results via natural language generation systems." Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. 2019.
- [110] Alsuwat, Emad, et al. "Adversarial data poisoning attacks against the PC learning algorithm." International Journal of General Systems 49.1 (2020): 3-31.
- [111] Alsuwat, Emad, et al. "Detecting adversarial attacks in the context of bayesian networks." IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Cham, 2019.
- [112] J. Steinhardt, P. W. Koh, and P. Liang, 'Certified Defenses for Data Poisoning Attacks', ArXiv170603691 Cs, Nov. 2017, Accessed: May 16, 2022. [Online]. Available: <http://arxiv.org/abs/1706.03691>
- [113] G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, 'Casting out Demons: Sanitizing Training Data for Anomaly Sensors', in 2008 IEEE Symposium on Security and Privacy (sp 2008), May 2008, pp. 81–95. doi: 10.1109/SP.2008.11.
- [114] J. Rauber, W. Brendel, and M. Bethge, 'Foolbox: A Python toolbox to benchmark the robustness of machine learning models', ArXiv170704131 Cs Stat, Mar. 2018, Accessed: Apr. 04, 2022. [Online]. Available: <http://arxiv.org/abs/1707.04131>
- [115] M.-I. Nicolae et al., 'Adversarial Robustness Toolbox v1.0.0', ArXiv180701069 Cs Stat, Nov. 2019, Accessed: Apr. 04, 2022. [Online]. Available: <http://arxiv.org/abs/1807.01069>
- [116] J. Lin, L. Dang, M. Rahouti, and K. Xiong, "ML Attack Models: Adversarial Attacks and Data Poisoning Attacks," *arXiv:2112.02797 [cs]*, Dec. 2021, Accessed: Feb. 17, 2022. [Online]. Available: <http://arxiv.org/abs/2112.02797>
- [117] A. Oseni, N. Moustafa, H. Janicke, P. Liu, Z. Tari, and A. Vasilakos, "Security and Privacy for Artificial Intelligence: Opportunities and Challenges," *arXiv:2102.04661 [cs]*, Feb. 2021, Accessed: Feb. 17, 2022. [Online]. Available: <http://arxiv.org/abs/2102.04661>
- [118] huybery, *VisualizingCNN*. 2022. Accessed: Jun. 28, 2022. [Online]. Available: <https://github.com/huybery/VisualizingCNN>
- [119] "Words matter: Alternatives for charged terminology in the computing profession." <https://www.acm.org/diversity-inclusion/words-matter> (accessed Jun. 28, 2022).

