Topic: HORIZON-MSCA-2022-SE-01-01
Call: HORIZON-MSCA-2022-SE-01
Type of Action: HORIZON-TMA-MSCA-SE

AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks

WP2 – Architecture Design, Requirements and Data

D2.1 Requirements and Reference Architecture

Editors	FOGUS
Authors	Dionysios Xenakis (FOGUS), Georgios Kalpaktsoglou (FOGUS), Athanasia Sampazioti (UPRC), Ignacio Lacalle Úbeda (UPV), Salvador Cuñat (UPV), José Alberto Martínez-Cadenas (UPV), Giorgio Bernardinetti (CNIT), Maria Niculae (BEIA), Zacharenia Lekka (K3Y), Nikolaos Sachpelidis Brozos (K3Y), Ilias Politis (ISI/ATHENA RC), Cristina Alcaraz (UMA), Javier Lopez (UMA), Hector Guzman (UMA), Stylianos Karagiannis (PDM), Luis Miguel Campos (PDM), Luis Landeiro Ribeiro (PDM), Leonidas Agathos (PDM), Paulo Correia (PDM), Daniel Grilo (PDM)
Dissemination Level	PU
Type	R
Version	2

Project Profile

Contract Number	101131292
Acronym	AIAS
Title	AI-ASsisted cybersecurity platform empowering SMEs to defend against adversarial AI attacks
Start Date	Jan 1st, 2024
Duration	48 Months

Partners

	UNIVERSITY OF PIRAEUS RESEARCH CENTER	EL
	BEIA CONSULT INTERNATIONAL SRL	RO
	UNIVERSIDAD DE MALAGA	ES
	K3Y	BG
	ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS	EL
	SUITE5 DATA INTELLIGENCE SOLUTIONS LIMITED	CY
	CONSORZIO NAZIONALE INTERUNIVERSITARIO PER LE TELECOMUNICAZIONI	IT
	FOGUS INNOVATIONS & SERVICES P.C	EL
	UNIVERSITAT POLITECNICA DE VALENCIA	ES
	PDM E FC PROJECTO DESENVOLVIMENTO MANUTENCAO FORMACAO E CONSULTADORIALDA	PT

Document History

VERSIONS

Table 1 Document history

Version	Date	Author	Remarks
0.1	19/7/2024	Dionysios Xenakis - FOG	ToC – initial version
0.2	5/8/2024	UPV	Methodology of requirements’ definition
0.3	25/9/2024	UPRC	Requirements of monitoring & analytics tool and technology gaps in the field of adversarial AI attack mitigation. Stakeholders
0.4	26/9/2024	UMA	Requirements of AIAS deception mechanism. Technology gaps in the field of deception mechanisms
0.5	27/9/2024	PDM	AIAS AI-driven Detection and Mitigation
0.6	30/09/2024	CNIT	User and Technical requirements of adversarial AI. Technology gaps of adversarial AI attack generation.
0.7	18/10/2024	FOG	Fix some formatting issues
0.8	30/10/2024	ISI	Section 4 intro; sections 4.1, 4.2 edited. Minor modifications throughout.
0.9	08/11/2024	UMA	Review and minimal changes in 2.2 Requirements
1.0	08/11/2024	UMA	Medium-interaction honeypots were added to Section 2.
1.1	28/11/2024	ISI	Requirement Methodology section updated regarding templates and definitions Completion of relevant user and technical requirements of the AIAS Mitigation mechanism.
1.2	6/12/2024	ALL	Review
1.3	20/12/2024	ALL	Refinement of the requirements
1.4	21/12/2024	UMA	Second review and corrections
1.5	27/12/2024	UPRC	Refinements of the architecture
2	31/12/2024	ALL	Final version

AIAS message

Executive Summary

The overall aims of the AIAS is to perform in-depth research on adversarial Artificial Intelligence (AI) designing and developing an innovative AI-based security platform for the protection of AI systems and AI-based operations of organisations, relying on Adversarial AI defence methods (e.g., adversarial training, adversarial AI attack detection), deception mechanisms (e.g., high interaction honeypots, digital twins, virtual personas) as well as on eXplainable AI (XAI) solutions that empower security teams to materialise the concept of “AI for Cybersecurity” (i.e., AI/Machine Learning (ML)-based tools to enhance the detection performance, defence and respond to attacks) and “Cybersecurity for AI” (i.e., protection of AI systems against adversarial AI attacks).

Deliverable 2.1 is dedicated to the documentation of the work done in the two tasks in Work Package (WP) 2, which are presented below as stated in the Grant Agreement.

- **Task 2.1: User, technical requirements and specifications**: This task is responsible to define the various types of users and stakeholders of the AIAS platform based on real users’ profiles with diverse expertise. Specify user-centred, technical and non-technical requirements for every module of AIAS and considers the operational needs of users and stakeholders by translating the user requirements into technical requirements.
- **Task 2.3: Architecture design**: This task will provide the abstract reference architecture of AIAS platform.

Table of Contents

Project Profile **2**
AIAS message **4**
Executive Summary **6**
Table of Contents **7**
Table of Figures **9**
Table of Tables **10**
1 Introduction **14**
1.1. Relationship to other deliverables **14**
1.2. Document structure **14**
2 Review of technology gaps **15**
2.1. Deception mechanisms **15**
2.1.1 State-Of-The-Art on Deception Technologies 16
2.1.2 Review of Technology Gaps in deception mechanisms 18
2.2. AI-based detection of cyberattacks and adversarial AI attacks **19**
2.2.1 State-of-the-Art in AI Detection Technologies 21
2.2.2 Emerging Trends and Future Directions 23
2.2.3 Gaps in Current AI-based Detection Technologies 24
2.3. Adversarial AI attack generation **26**
2.4. Adversarial AI attack mitigation **27**
2.4.1 Identification of Technological Gaps 28
2.4.2 Addressing Technological Gaps in Adversarial AI Mitigation in AIAS 28
2.5. Security data fusion **30**
3 Stakeholders **43**
4 User and Technical Requirements **44**
4.1. Methodology **44**
4.2. Definition of a requirement **44**
4.3. User, Functional & non- Functional requirements of AIAS modules **51**
4.3.1 AIAS Deception mechanism 51
4.3.2 AIAS Detection mechanism 58

4.3.3 AIAS Adversarial AI (Weaponizer) 62
4.3.4 AIAS Mitigation mechanism 66
4.3.5 AIAS Security Data Fusion 74
4.3.6 Monitoring and analytics 79
5 Reference Architecture 84
5.1. AIAS Architecture principles 84
5.2. Proactive Defence through Adversarial Simulation 85
5.3. Layered Defence with Deception Technologies 85
5.4. Continuous Learning and Adaptation 85
5.5. Explainability and Human-Centric Decision Support 86
5.6. Secure, Decentralized, and Collaborative Data Management 86
5.7. Scalability and Modularity for Adaptability 87
5.8. AIAS Architecture description 87
5.8.1 AIAS Adversarial AI Engine and Deception 88
5.8.2 AIAS AI-driven Detection and Mitigation 91
6 Conclusions 94
References 95

Table of Figures

Fig. 1 Conceptual map of research lines and gaps for Data Fusion Module 31
Fig. 2 The requirements capture procedure 46
Fig. 3 Initial design of AIAS AI-based Data Fusion Module for detection.....87
Fig. 4 Architecture of the Adversarial AI Engine Module 88
Fig. 5 Architecture of the Deception Layer 89
Fig. 6 AI-Driven Detection and Mitigation Architecture 92

Table of Tables

Table 1 Document history....4
Table 2 Abbreviation Table....11
Table 3 Comparison Table....18
Table 4 Technology gaps in deception mechanisms....19
Table 5 De-facto standard cyber-attacks representation and modelling attempts....32
Table 6 AIAS key stakeholder and their benefit....43
Table 7 Template for user requirements....48
Table 8 Template for technical requirements....49
Table 9 Functional and non-functional requirements of the AIAS deception mechanism....51
Table 10 User requirements of the AIAS Deception mechanism....57
Table 11 Functional and non-functional requirements of the AIAS Detection mechanism....58
Table 12 User requirements of the AIAS Detection mechanism....61
Table 13 Functional and non-functional requirements of the AIAS adversarial AI component....62
Table 14 User requirements of the AIAS adversarial AI component....66
Table 15 User requirements of the AIAS mitigation mechanism....67
Table 16 Functional and non-functional requirements of the AIAS mitigation mechanism....68
Table 17 User requirements of the AIAS Security Data Fusion component....75
Table 18 Functional and non-functional requirements of the AIAS Security Data Fusion component....75
Table 19 Functional and non-functional requirements of the AIAS monitoring and analytics tool....79
Table 20 User requirements of the AIAS monitoring and analytics tool....83

Table 2 Abbreviation Table

Abbreviation	Description
UPnP	Universal Plug and Play Protocol
SSH	Secure Shell
RFI	Remote File Inclusion
MHN	Modern Honey Network
ICS	Industrial Control System
ICT	Information Communication Technology
IoT	Internet of Things
IoV	Internet of Vehicles
AI	Artificial Intelligence
XAI	Explainable AI
API	Application Programming Interface
SQL	Structured Query Language
ML	Machine Learning
SME	Small and Medium-sized Enterprise
GMM	Gaussian Mixture Models
PCA	Principal Component Analysis
KNN	K-Nearest Neighbours
LIME	Local Interpretable Model-Agnostic Explanations
AIDM	AI-based Detection Module
LLRL	LifeLong Reinforcement Learning
SHAP	SHapley Additive exPlanations
IPFS	InterPlanetary File System
GDPR	General Data Protection Regulation
MTTR	Mean Time To Recovery
GAN	Generative Adversarial Network
TTPs	Tactics, Techniques, and Procedures
AI2EM	Adversarial AI Engine Module
DNN	Deep Neural Network
DoS	Denial of Service
IDS	Intrusion Detection System
UEBA	User and Entity Behavior Analytics
IDPS	Intrusion Detection and Prevention Systems

SIEM	Security Information and Event Management
UI/UX	User Interface/ User Experience
AAA	Authorization, And Accounting
OSS	Open-Source Software
RNN	Recurrent Neural Network
NLP	Natural Language Processing
URL	Uniform Resource Locator
HTML	HyperText Markup Language
XML	Extensible Markup Language
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart
TOR	The Onion Routing
I2P	Invisible Internet Project
CTI	Cyber Threat Intelligence
OSINT	Open-Source Intelligence
SOC	Security Operations Center
MTD	Moving Target Defence
CID	Content IDentifier
XSS	Cross Site Scripting
RDF	Resource Description Framework
MITRE ATT&CK	MITRE Adversarial Tactics, Techniques and Common Knowledge
STIX	Structured Threat Information Expression
CAPEC	Common Attack Pattern Enumeration and Classification
CVE	Common Vulnerabilities and Exposures
CWE	Common Weakness Enumeration
PCAP	Packet CAPture
CICIDS	Canadian Institute for Cybersecurity - Intrusion Detection Evaluation Dataset
XPath	XML Path Language
LSTM	Long Short-Term Memory
JSON	JavaScript Object Notation
XML	Extensible Markup Language
YAML	Yet Another Markup Language
AJAX	Asynchronous JavaScript and XML
SQL	Structured Query Language

1 Introduction

This deliverable entitled “*Requirements and Reference Architecture*” is responsible for defining the AIAS requirements and its reference architecture. Overall, the present deliverable has the following objectives:

- Describe user, technical requirements and specifications of the AIAS platform [AIA].
- Define the design architecture of the AIAS platform.

1.1. Relationship to other deliverables

This section describes the relationship between D2.1 and the upcoming technical deliverables:

- **D2.2 Specifications & Business Cases**: The defined uses case that will evaluate the platform will comply with the user, functional and non-functional requirements defined in D2.1.
- **D3.1 AIAS Deception Layer**: The D3.1 deliverable will include the AIAS deception layer and the monitoring and analytics tools, these components will be design and developed according to the requirements defined in the current document.
- **D3.3 Adversarial AI Engine**: The delivered Adversarial AI Engine will be designed and developed in accordance with the corresponding requirements in D2.1.
- **D4.1 AI-based Detection of Adversarial**: The delivered AI-based detection module will be designed and developed complying with the corresponding requirements documented into D2.1.
- **D4.2 Mitigation of Adversarial AI Attacks & XAI**: The delivered mitigation component will be designed and developed according with the corresponding D2.1 requirements.
- **D5.1 Platform Integration**: The components will be assessed and refined according to the requirements defined in this deliverable.
- **D5.2 Platform Evaluation**: The overall platform assessment will occur according to the requirements defined in D2.1.

1.2. Document structure

The rest of the document is organized as follows:

- **Section 2** highlights the main identified technology gaps in the research areas where the AIAS projects lies in including the areas of the deception mechanisms, AI-based detection mechanisms, adversarial AI attack generation and mitigation, as well as, in the security data fusion.
- **Section 3** defines the key stakeholders of the AIAS platforms and how they will benefit from the platform.
- **Section 4** includes the methodology for defining the requirements within the AIAS ecosystem, together with the requirements (user, functional and non-functional) for the AIAS’ components.
- **Section 5** describes the architectural principles used within the AIAS platform together with a detailed technical analysis of the AIAS’ components.
- **Section 6** outlines the conclusions of the current deliverable.

2 Review of technology gaps

This Section describes the technology gaps identified in the areas where the AIAS platform lies in.

2.1. Deception mechanisms

To study the current technological gaps in deception technologies, we developed a classification system for honeypots based on key characteristics identified from [AJF]. This classification also aims to identify which implementations best suit our needs, enabling deeper exploration and highlighting the most important features of each.

1. Based on the implementation:
   - **Virtual:** These honeypots are deployed on a virtualized environment, such as virtual machines or containers, allowing them to run multiple instances on a single server. They do not require dedicated physical hardware, making them cost-effective and easy to scale.
   - **Physical:** Honeypots implemented on a dedicated machine with its own physical hardware and unique IP address. These honeypots provide a more realistic environment, as they replicate a real system in both functionality and network presence, but the resource cost is higher.
2. Based on the level of interaction:
   - **Low-interaction:** Designed to simulate basic system functionality with limited interaction, these honeypots focus on detecting attacks while minimizing risk. As they do not offer attackers many opportunities to interact with the system, they are less resource-draining and safer to deploy.
   - **Medium-interaction:** These honeypots offer more interaction than low-interaction honeypots but do not fully replicate real systems as high-interaction honeypots. Medium-interaction honeypots emulate selected system behaviours and services to engage attackers moderately, capturing useful information while maintaining a lower risk profile.
   - **High-interaction:** These honeypots replicate real systems and services in great detail, allowing attackers to interact extensively with the system. High-interaction honeypots aim to engage attackers for longer periods, providing detailed information about the attacks, but they are also riskier since attackers may have more opportunities to exploit vulnerabilities.
3. Based on the purpose:
   - **Research:** Honeypots that are primarily used to gather information about attackers, their methods, and their tools. These honeypots are typically employed in environments where the aim is to study cyberattacks in detail, with a focus on learning and analysis, rather than active defence.
   - **Production:** These honeypots are deployed with the aim of diverting attackers from real systems. They simulate real system activities and services to deceive attackers, wasting their time and protecting critical assets by redirecting malicious activity to the honeypot.
4. Based on the activity:
   - **Passive:** These honeypots focus on quietly gathering information from attackers without actively responding. Their main function is to monitor and log attacker behaviour for analysis, providing valuable intelligence.
   - **Active:** In contrast, active honeypots engage attackers in real-time, interacting with them to distract

and divert attention away from critical systems. They may simulate system responses, alter attacker activities, and even guide the attacker through a deceptive environment, providing a more proactive defence strategy.

5. Based on the uniformity:
   - **Homogeneous:** Honeypots that deploy a single type of decoy or trap to deceive attackers. Homogeneous honeypots may only be effective against certain types of attacks, limiting their versatility in detecting a wide range of threats.
   - **Heterogeneous:** These honeypots implement multiple types of decoys, traps, and security tools, creating a more complex and varied environment. This approach increases their ability to detect and respond to different types of attacks, as attackers face multiple layers of deception.
6. Based on the actions taken:
   - **Static:** Honeypots with fixed configurations that remain the same regardless of the attack. Static honeypots always behave in the same way, providing consistent responses. While simpler to manage, they may be easier for attackers to recognize after repeated interactions.
   - **Dynamic:** These honeypots can adapt to changes and modify their behaviour based on the attacker's actions or past activity. They dynamically alter their responses, simulating a more realistic system, which makes it harder for attackers to recognize the deception and allows for more effective engagement.

2.1.1 State-Of-The-Art on Deception Technologies

The following section presents a state-of-the-art overview of deception technologies, offering insights into the current landscape of honeypot systems and identifying potential gaps that could be addressed through the exploration of new approaches. Below a list of honeypot technologies as presented in the literature is provided while a table for each honeypot indicates the respective attributes based on the classification described in Subsection 2.1.

- **HoneyPy [HPY].** HoneyPy is a low-interaction honeypot for the monitoring of network threats. The tool is easily customizable by adding plug-ins. However, it is no longer in active development.

Virtual	Low-interaction	Research	Passive	Heterogeneous	Static

- **HoneyBot [IFL].** HoneyBot is a honeypot specifically designed for robotic systems. It consists in simulating unsafe actions when physically performing safe actions to fool adversaries into believing their actions are successful. At the same time, all the communication is logged for attacker attribution and threat model creation.

Virtual	High-interaction	Research and Production	Active	Homogeneous	Static

- **HoneyC [SWK].** HoneyC is a platform independent framework for low-interaction honeypot designed to address some of the shortcomings of traditional high-interaction client honeypots.

Virtual	Low-interaction	Research	Passive	Homogeneous	Static

- **IoTPOT [RRM].** IoTPOT is a medium-interaction honeypot that emulates Telnet services of IoT devices to analyse attacks. It provides functionality for scanning the attacking IP addresses.

Virtual	Medium-interaction	Research and Production	Passive	Homogeneous	Static

- **U-PoT [HAU].** U-PoT is a honeypot for UPnP (Universal Plug and Play Protocol) that solves the problem of existing low-interaction honeypots while providing the advantages of high-interaction honeypots. The honeypot is capable of emulating a device by creating a snapshot of it after invoking all the actions of its services. Then, the honeypot is ready to listen incoming requests and return states properly to deceive attackers.

Virtual	High-interaction	Production	Passive	Homogeneous	Static

- **Cowrie [COW].** Cowrie is a medium-interaction SSH (Secure Shell) and Telnet honeypot designed to log brute force attacks and the shell interaction performed by the attacker. Additionally, Cowrie can run as a proxy, logging the activity to other systems.

Virtual	Medium-interaction	Research	Passive	Homogeneous	Static

- **Dionaea [DIO1, DIO2].** Dioanea is a low-interaction honeypot that emulates network services to gather information about how malware exploits vulnerabilities exposed by those services.

Virtual	Low-interaction	Research and Production	Passive	Homogeneous	Static

- **Glastopf [GLA1, GLA2].** Glastopf is a low-interaction honeypot for web applications capable of emulating thousands of web vulnerabilities such as Remote File Inclusion (RFI) or Structured Query Language (SQL) injection.

Virtual	Low-interaction	Research and Production	Passive	Heterogeneous	Static

- **Modern Honey Network (MHN) [MHN].** MHN is a user-friendly centralized sever for data collection of honeypots. It includes several honeypot technologies such as Cowrie, Dioanea, and Glastopf, among others.

Virtual	Low-interaction	Research and Production	Passive	Heterogeneous	Static

- **HoneyDrive [HDR].** HoneyDrive is a virtual appliance with a Linux distro installed that contains pre-configured honeypot software packages such as Kippo SSH honeypot, Dionaea and Amun malware honeypots, Honeyd low-interaction honeypot, Glastopf web honeypot and Wordpot, Conpot ICS (Industrial Control System) honeypot, Thug and PhoneyC honey clients and more.

Virtual	Low-interaction	Research and Production	Active	Heterogeneous	Static

HoneyThing [HTH]. HoneyThing is a low-interaction honeypot for the Internet of Things (IoT) that emulates the TR-069 protocol. It simulates some popular vulnerabilities for RomPager (embedded web server found in many IoT devices).

Virtual	Low-interaction	Research	Passive	Homogeneous	Static

- **Conpot [CON].** Conpot is a low-interaction server-side ICS honeypot capable of simulating a wide range of industrial protocols to emulate complex infrastructures to convince adversaries that they are in huge industrial environments.

Virtual	Low-interaction	Research and Production	Passive	Homogeneous	Static

- **Kippo [KIP].** Kippo is a medium interaction SSH honeypot designed to log brute force attacks and, most importantly, the entire shell interaction performed by the attacker.

Virtual	Medium-interaction	Research	Passive	Heterogeneous	Static

- **HonSSH [HSS].** HonSSH is a tool designed to be used with a high-interaction honeypot. HonSSH sits between the attacker and the honeypot by creating two separate SSH connections. This project was inspired by Kippo and has made use of its logging and interaction mechanisms.

Virtual	High-interaction	Research	Passive	Homogeneous	Static

Table 3 Comparison Table

Technology	Implementation	Interaction	Purpose	Activity	Uniformity	Action
HoneyPy	Virtual	Low-interaction	Research	Passive	Heterogeneous	Static
HoneyBot	Virtual	High-interaction	Both	Active	Homogeneous	Static
HoneyC	Virtual	Low-interaction	Research	Passive	Homogeneous	Static
IoTPOT	Virtual	Medium-interaction	Both	Passive	Homogeneous	Static
U-PoT	Virtual	High-interaction	Production	Passive	Homogeneous	Static
Cowrie	Virtual	Medium-interaction	Research	Passive	Homogeneous	Static
Dioanea	Virtual	Low-interaction	Both	Passive	Homogeneous	Static
Glastopf	Virtual	Low-interaction	Both	Passive	Heterogeneous	Static
MHN	Virtual	Low-interaction	Both	Passive	Heterogeneous	Static
HoneyDrive	Virtual	Low-interaction	Both	Active	Heterogeneous	Static
HoneyThing	Virtual	Low-interaction	Research	Passive	Homogeneous	Static
Conpot	Virtual	Low-interaction	Both	Passive	Homogeneous	Static
Kippo	Virtual	Medium-interaction	Research	Passive	Heterogeneous	Static
HonSSH	Virtual	High-interaction	Research	Passive	Homogeneous	Static

Table 3 presents a variety of virtual honeypot technologies that predominantly feature static configurations, meaning they do not adapt dynamically to attacks. Most of the technologies focus on low-interaction levels, with a few high-interaction honeypots to engage attackers extensively. They are mainly designed for research purposes, with a smaller subset of production tools. Almost all honeypots follow a passive approach, prioritizing data collection over active defence and the majority employ homogeneous decoys, limiting their scope of deception.

2.1.2 Review of Technology Gaps in deception mechanisms

Analysing the table above we can find certain gaps in the state of the art of deception mechanisms. Table 4 presents these technology gaps and matches them with the specific needs.

Table 4 Technology gaps in deception mechanisms

Short Description	Gap	Need
Low dynamicity in interaction and adaption	Those honeypots classified as static or passive lack on real-time adaption to attackers’ tactics. Their static nature restricts the ability to update the deception strategy.	Use of more **AI** or **machine learning** tools such as **reinforcement learning** to create honeypots with real-time response adjustment based on attackers’ behaviour.
Limited number of physical interaction honeypots	While the majority of honeypots are virtual, those focused on physical systems such as industrial IoT are less common and usually too specific.	More high-interaction physical honeypots for specific environments are needed to protect these areas.
Lack of heterogeneous honeypots	Most of the honeypots are homogeneous, being easier to detect and gathering less information than heterogeneous honeypots, which increase the deceptive power.	There is a need of more heterogeneous honeypots capable of mimicking a wide range of devices.
Narrow subset of attacks	Many honeypots focus on a narrow subset of attacks.	Developing deception systems capable to respond to a wider variety of attack vectors.
Scalability	Most honeypots are limited in terms of scalability and geographical distribution.	Scalable honeypots that can operate in distributed networks, particularly for large networks such as Internet of Vehicles (IoV) systems.
Isolated solutions	Current honeypots are too focused on specific domains without considering cross-domain interactions.	Honeypots that bridge different domains such as transport, smart devices, energy

Addressing these gaps will require the development of more sophisticated honeypots. Nonetheless, the current state of the art serves as a starting point where each technology studied brings interesting features that may lead to a powerful deception tool. To bridge these gaps, potential solutions include incorporating AI models to enhance adaptability and response to evolving threats, developing heterogeneous tools capable of addressing a wider range of attack vectors, and emphasising on the scalability of the technologies to handle larger environments. Additionally, overcoming the cross-domain gap by creating tools that can integrate and link different fields will be crucial for a more flexible deception system.

2.2. AI-based detection of cyberattacks and adversarial AI attacks

The concept of detection mechanisms is critical in the modern cybersecurity landscape, particularly when addressing adversarial AI attacks. With the rapid advancement of AI systems, attackers have begun to exploit vulnerabilities in AI models, targeting them through methods like data poisoning, adversarial examples, and model evasion. Detection mechanisms play a vital role in identifying these malicious activities and ensuring the integrity and security of AI systems [SAH].

Detection mechanisms in AI-driven environments involve monitoring data inputs and system behaviour to identify anomalies that indicate the presence of an attack. These mechanisms can be deployed in various stages of the AI system lifecycle, from data acquisition and preprocessing to model training and inference [SAH]. Artificial Intelligence (AI) and Machine Learning (ML) systems are becoming increasingly integral to various business operations, particularly in sectors such as finance, healthcare, smart cities, and autonomous systems. However, the widespread adoption of AI also brings unique challenges, especially in terms of security. One of the most critical challenges is the growing threat of adversarial AI attacks - deliberate efforts by malicious actors to manipulate or corrupt AI models, leading to inaccurate outputs or system failures. Adversarial AI attacks can take many forms, including data poisoning, adversarial perturbations, model evasion, and exploitation of system vulnerabilities, all of which can severely impact the functionality, reliability, and security of AI-driven systems [GBA].

In this context, detection mechanisms play a pivotal role in safeguarding AI systems against adversarial threats. The goal of detection mechanisms is to identify and mitigate attacks in real-time or as early as possible before they can cause significant harm to the AI models or the larger system. Unlike traditional security mechanisms, adversarial detection systems must be capable of: (i) Monitoring AI models continuously, (ii) detecting anomalies in data and system behaviours, and (iii) identifying adversarial inputs designed to deceive the AI [RKZ].

This is particularly crucial in Small and Medium-sized Enterprises (SMEs), which may not have the extensive resources required to develop custom security solutions but are increasingly adopting AI to remain competitive. A robust detection framework must provide (i) real-time monitoring, (ii) automated response capabilities, and (iii) an integrated approach that works with deception mechanisms (such as honeypots and virtual personas) to form a multi-layered defence strategy [VAJ].

Detection mechanisms are intended to act as the first line of defence against adversarial attacks within the AIAS platform. These mechanisms continuously monitor the behaviour of the AI system, identifying unusual patterns and potential threats for detailed analysis or automated mitigation. Such mechanisms are critical across all phases of the AI lifecycle, including training, deployment, and operational stages, ensuring comprehensive protection [CJH]. The main objectives of detection mechanisms in AIAS include:

- **Early Identification of Adversarial Inputs:** Detection of malicious inputs at several levels, either during the training phase of an AI model or during usage in a productive environment.
- **Real-time Threat Detection:** Continuously monitoring data streams and system behaviour to spot possible signs of adversarial attacks as they occur.
- **Minimising False Positives and False Negatives:** Ensuring that legitimate anomalies, like unexpected but valid inputs, aren’t mistakenly flagged as attacks, while keeping the chances of missing real threats to a minimum.
- **Supporting Incident Response:** Allowing security teams to act quickly and effectively by providing insights and explanations using Explainable AI to better understand the anomalies detected.
- **Integration with Deception Tools:** Using deception mechanisms such as honeypots and digital twins to trick attackers into revealing their tactics, thereby enhancing detection accuracy.

Through these objectives, the AIAS platform’s detection mechanisms ensure that AI systems are resilient to adversarial attacks and that SMEs are equipped with state-of-the-art tools to defend their AI operations from emerging cybersecurity threats.

Detection mechanisms not only provide security during operational stages but also during the AI system’s training and deployment phases [ZZA]. For instance, poisoning attacks [TZC], where malicious data is injected during training, can severely compromise the accuracy of the resulting trained model—the final model that is deployed for real-world use. Detection mechanisms can identify these issues early, during the training phase, to preserve the integrity and reliability of the model before it is operationalized. The work is going to expand, in the ensuing sections, on the current technologies of detection mechanisms, state gaps in these technologies, and enumerate particular requirements that will need to be met for the AIAS platform to effectively detect adversarial AI attacks.

2.2.1 State-of-the-Art in AI Detection Technologies

The detection of adversarial attacks on AI systems is a rapidly evolving area of research, driven by the increasing complexity of both attacks and defence mechanisms. Adversarial AI attacks, such as data poisoning and adversarial perturbations [CJH], exploit weaknesses in machine learning models, leading to compromised outputs that could have disastrous consequences in sensitive areas such as finance, healthcare, and autonomous systems. Detection technologies have been developed to identify such attacks before they can cause significant damage, but the field is still in its infancy and faces numerous challenges.

One of the most significant challenges in this field is the adaptive nature of adversarial attacks. Attackers continuously develop new techniques to bypass existing detection mechanisms, rendering static defence strategies ineffective over time. This dynamic adversarial landscape necessitates the development of adaptive and robust detection systems capable of evolving alongside emerging threats. For instance, a survey on adversarial attacks and defences in machine learning-powered networks highlights the rapid evolution of attack methods and the corresponding need for adaptive defence strategies [WYS].

Anomaly detection is widely used in various fields, including cybersecurity, for identifying inputs and behaviours that deviate significantly from the norm. In the context of AI security, anomaly detection involves monitoring inputs and system behaviours for signs of adversarial activity, such as unexpected variations in data patterns or model predictions. The most common approaches include:

- **Statistical Models:** Techniques such as Gaussian Mixture Models (GMM) and Principal Component Analysis (PCA) are used to model normal data distributions, identifying anomalies based on deviations from these distributions. While effective in many cases, statistical models can struggle to detect adversarial samples that closely mimic legitimate inputs [AGS].
- **Distance-based Methods:** These methods, such as *k*-Nearest Neighbours (KNN) and clustering techniques, measure the distance between data points to identify outliers. Inputs that are far from the majority of the data are flagged as anomalies. However, sophisticated adversarial attacks often remain within the distribution of legitimate data, making them harder to detect using this method [HRK].

- **Machine Learning Approaches:** MLmodels, including autoencoders and clustering algorithms, have been developed to detect anomalies by learning the underlying structure of normal data during training and flagging deviations during operation. However, these models can be computationally expensive and may suffer from high false-positive rates, making them difficult to implement in real-time systems [NAB].

The limitations of traditional anomaly detection methods include the challenge of identifying adversarial samples, which are often crafted to closely resemble legitimate inputs, making detection difficult [TRF]. Additionally, anomaly detection techniques can produce a high volume of false positives, resulting in alert fatigue for security teams [ABA]. The computational overhead of these methods further restricts their scalability in real-time applications.

Adversarial sample detection specifically targets inputs that have been manipulated to deceive AI models. These techniques are designed to recognize subtle, often imperceptible, modifications made to data that can drastically affect model predictions.

- **Gradient-based Methods**: Gradient-based techniques analyse the gradients of the loss function with respect to the input data. Adversarial examples are typically generated by exploiting gradients to maximise model errors, making this a promising method for detecting adversarial inputs. However, these methods can be computationally expensive and require access to the model’s internal workings, which might not always be feasible [DMS].
- **Perturbation-based Detection**: This method identifies adversarial samples by measuring how small perturbations in the input affect the model’s output. Adversarial inputs are often highly sensitive to such perturbations, resulting in noticeable changes in predictions [IMK].
- **Feature Squeezing**: Feature squeezing reduces the precision of input data (e.g., by lowering the colour depth of an image) and observes how the model’s prediction changes. If a significant change occurs, it may indicate an adversarial sample. While effective in some cases, adversarial examples can be crafted to evade feature squeezing [LYG].

The strengths of these techniques lie in their specific design for detecting adversarial inputs, making them more effective than general anomaly detection methods. They are capable of identifying subtle perturbations that may be imperceptible to humans yet significantly impact ML models. However, their limitations include a high computational cost, which hinders real-time deployment. Additionally, adversaries constantly evolve their tactics to bypass these methods, and balancing sensitivity with specificity remains challenging; achieving high detection rates often leads to an increase in false positives.

Rather than focusing solely on inputs, model behaviour monitoring tracks changes in the AI system’s internal decision-making processes. This method aims to identify adversarial attacks by analysing how the model’s behaviour deviates from its normal operational patterns.

- **Prediction Confidence Analysis**: This technique monitors the confidence levels of the model’s predictions. Adversarial attacks often cause erratic confidence shifts, which can serve as a red flag for potential manipulations [GAH].
- **Decision Boundary Monitoring**: By continuously analysing how close inputs are to decision

boundaries, this method detects inputs that attempt to exploit weaknesses in the model’s classification rules. Adversarial inputs often lie close to the decision boundaries, making them easier to spot using this technique [HOT].

- **Explainable AI (XAI)**: Explainable AI tools, such as LIME (Local Interpretable Model-Agnostic Explanations) [PSR] and SHAP (SHapley Additive exPlanations) [NYM], provide transparency into the model’s decision-making process. By offering a deeper understanding of why a model made a certain decision, XAI can help detect unusual behaviours caused by adversarial inputs.

The strengths of model behaviour monitoring include its ability to detect a broader range of attacks by analysing the AI system’s internal operations rather than focusing solely on input data. This approach offers security teams detailed insights into the model’s decision-making processes, enabling a deeper understanding and more effective responses to potential attacks. However, these methods are resource-intensive, requiring extensive monitoring infrastructure, and may introduce latency, which can impact real-time decision-making systems. Additionally, false positives remain a risk, as deviations in model behaviour can occur for reasons unrelated to adversarial attacks.

2.2.2 Emerging Trends and Future Directions

Recent advancements in AI detection technologies are pushing the boundaries of what is possible in adversarial attack detection. Several emerging trends offer promising directions for future research and development:

- **Hybrid Detection Models**: Combining multiple detection approaches, such as anomaly detection, adversarial sample detection, and behaviour monitoring, into a single framework improves detection accuracy and reduces false positives. These hybrid models leverage the strengths of each method to create a more comprehensive security solution [JNT].
- **AI-driven Cybersecurity Systems**: Reinforcement learning, and semi-supervised learning models are increasingly being applied to AI security, enabling systems to dynamically adapt to new types of adversarial attacks. These approaches are designed to learn from attack patterns and evolve their detection mechanisms over time [SIH].
- **Deception-based Detection**: Incorporating deception mechanisms such as honeypots [IND] and digital twins [SAK] into detection strategies is an emerging field. These tools lure adversaries into revealing their tactics, allowing detection systems to gather crucial intelligence about the attack without compromising real systems.

While state-of-the-art AI detection technologies offer powerful tools for identifying adversarial attacks, they still face several challenges, particularly in real-time and resource-constrained environments. The AIAS platform aims to overcome these limitations by integrating advanced detection techniques with deception-based mechanisms, creating a more robust and scalable solution tailored to SMEs. Through these advancements, the platform will significantly enhance the ability to detect, analyse, and respond to adversarial AI threats.

2.2.3 Gaps in Current AI-based Detection Technologies

Despite significant advancements in AI detection technologies [PVF], numerous gaps persist, particularly in defending against advanced adversarial attacks. These include evasion attacks [KOV], where malicious inputs are crafted to bypass detection systems without altering the underlying functionality, and poisoning attacks [TZC], which compromise model integrity by introducing manipulated data during the training phase. Additionally, sophisticated techniques such as transferability attacks [WFN], where adversarial examples designed for one model successfully deceive others, and model inversion attacks [FHQ], which extract sensitive information from AI systems, highlight the evolving nature and complexity of adversarial threats. These gaps represent both technological limitations and challenges in practical implementation, especially for SMEs that lack the resources and expertise of larger enterprises. Below are some of the key gaps in current detection technologies that must be addressed to build a robust defence system for AI-driven operations.

One of the primary challenges in current detection systems is their limited awareness of adversarial techniques. Many detection tools are optimised for traditional attack vectors (e.g., malware or phishing) and are not designed to recognize the unique and subtle manipulations involved in adversarial attacks on AI systems. These attacks often involve imperceptible changes to input data, such as adversarial perturbations or data poisoning, that can completely alter the output of a ML model [CAA].

This gap exists because current detection methods often struggle to capture subtle perturbations, allowing adversaries to bypass detection mechanisms. For instance, adversarial samples crafted to manipulate neural networks frequently evade traditional anomaly detection approaches by falling within acceptable data distributions. Moreover, adversarial techniques evolve rapidly, with new methods continually emerging to target specific vulnerabilities in AI models. As detection systems are not consistently updated to incorporate these evolving attack strategies, they risk becoming increasingly ineffective over time [MAL].

Real-time monitoring is crucial for detecting adversarial attacks as they unfold, but many existing detection technologies are limited by their inability to operate in realtime. Current anomaly detection systems, for example, may require batch processing or periodic evaluation of system logs, leading to delays in identifying and responding to attacks [SAF].

This gap exists because AI-based systems, especially those in sensitive and critical environments such as finance or healthcare, require instant detection and response capabilities. Any delay in identifying adversarial attacks can lead to significant damage or data integrity loss, with potential risks to human lives or financial stability. Although advanced detection methods, such as perturbation-based detection, offer high accuracy, they often demand substantial computational resources, limiting their use in real-time applications. This challenge is particularly acute for SMEs, which may lack the infrastructure to support such resource-intensive operations.

Many state-of-the-art detection mechanisms struggle to scale effectively, particularly in complex, real-world environments with large amounts of data. As AI systems grow more complex and are deployed across various domains (e.g., IoT networks, autonomous systems, and cloud-based environments), detection mechanisms must keep up with the increased volume and variety of data [ATA].

This gap arises because detection systems often face a trade-off between accuracy and computational

efficiency [CAF]. High-accuracy systems not only tend to be computationally expensive but also require extensive, high-quality datasets to perform effectively. This 'data hunger' poses significant challenges, particularly for SMEs, which may lack access to such datasets or the resources to curate them [OLJ]. Additionally, these companies struggle with the costs and infrastructure required to scale detection systems effectively. SMEs need detection solutions that are lightweight, scalable, and capable of achieving high accuracy without excessive resource demands or reliance on large datasets [AKM]. The balance between false positives (incorrectly flagging legitimate activity as malicious) and false negatives (failing to detect an actual attack) is a significant challenge in AI detection systems [AAS]. High rates of false positives can overwhelm security teams and lead to "alert fatigue" (overwhelming number of alerts desensitizes the people tasked with responding to them, leading to missed or ignored alerts or delayed responses), where critical threats may be ignored or missed. On the other hand, false negatives leave the system vulnerable to undetected adversarial attacks [BTS]. This gap exists because most current detection systems struggle to balance accuracy, often leaning toward one of two extremes. Systems focused on anomaly detection may generate excessive false positives due to real-world data variability, while adversarial detection systems can be circumvented if attackers create inputs that closely resemble legitimate data. This challenge is especially problematic in AI-based systems, where the boundary between legitimate and adversarial inputs is often blurred, complicating the design of detection mechanisms that consistently perform well across diverse environments.

Although deception mechanisms (such as honeypots and digital twins) have proven effective in cybersecurity for luring attackers into interacting with false systems, many current detection technologies do not integrate these mechanisms into their architectures [EHM]. The combination of detection and deception could provide an additional layer of security, allowing detection systems to gather valuable intelligence on attack methods without exposing real assets. This gap exists because integrating detection and deception could significantly improve threat identification accuracy by observing adversarial behaviour within decoy systems. However, most detection systems are not designed to leverage this interaction, missing an opportunity to enhance detection robustness. Additionally, deception mechanisms can help reduce false positives by isolating suspicious activities within decoy environments, allowing for more precise analysis of potential threats [ZLT].

Explainability, or the ability to understand why a system flagged a particular input as adversarial, is still underdeveloped in many detection mechanisms. EXplainable AI (XAI) tools such as LIME and SHAP [GPD] have shown promise in providing insights into model behaviour, but they are not yet widely adopted in detection technologies. This gap arises because, without explainability, security teams may find it challenging to understand why certain inputs are flagged as adversarial, making it difficult to take corrective actions or refine the detection system. This lack of transparency reduces trust in the system and can lead to critical threats being overlooked or ignored. Explainable AI tools could mitigate false positives by providing insights into the reasoning behind detection decisions, enabling teams to fine-tune the system for improved accuracy [CBJ].

Current detection technologies, while advanced in many respects, still exhibit significant gaps that must be addressed to provide robust protection against adversarial attacks. These gaps—ranging from real-time monitoring challenges to limited integration with deception and explainability tools—underscore the need for further research and development. The AIAS platform aims to close these gaps by introducing scalable, real-time detection mechanisms that are both resource-efficient and tightly integrated with deception strategies,

making them particularly suited for SMEs.

2.3. Adversarial AI attack generation

Adversarial AI attack generation remains a rapidly evolving field, but there are still significant technological gaps that limit the effectiveness, scalability, and real-world application of these methods. Identifying these gaps is crucial for advancing both adversarial capabilities and defensive measures in AI systems.

1. Limited Real-World Applicability of Generated Attacks.

   Most adversarial attacks are developed and tested in controlled environments using benchmark datasets like MNIST [MNT] or ImageNet [INET]. However, translating these attacks to real-world scenarios presents major challenges. Variability in data distributions, environmental factors, and non-differentiable operations in practical systems make effective attack generation far more complex. Creating adversarial methods that can generalize and function reliably beyond synthetic benchmarks remains an unresolved issue in this domain.

2. Lack of Understanding of Black-Box Model Vulnerabilities.

   Real-world AI systems often operate as black-box models, where internal details such as parameters and gradients are inaccessible. While approaches like transfer attacks and query-based methods exist for black-box settings, they are significantly less reliable and efficient compared to white-box attacks. These techniques often require a large number of queries, leading to increased costs and higher risk of detection. A key technological gap is the ability to generate strong adversarial examples against black-box models while minimizing resource usage and maintaining stealth.

3. Adversarial Attack Automation and Tooling.

   Current adversarial attack generation tools lack scalability and ease of use. Many available tools are limited in scope, difficult to configure, and require specialized expertise, which limits their broader adoption by researchers and security professionals. The absence of user-friendly, automated, and comprehensive frameworks poses a barrier to effective testing of AI model robustness. Addressing this gap requires developing standardized tools that can simplify the attack generation process and make it more accessible.

4. Limited Effectiveness Against “Adversarial”-Trained Models

   Adversarial training has proven to be a robust defence mechanism, as models trained on adversarial examples often exhibit enhanced resistance. However, generating attacks that can effectively bypass these defences is still a significant challenge. Current methods either fail against “adversarial”-trained models or require large perturbations, making the attacks more detectable. Developing new attack techniques capable of breaking through these improved defences without compromising subtlety remains a critical technological gap.

5. Lack of Transferability Insights Across Diverse Model Architectures

   The ability of adversarial examples to transfer between different models is a fundamental aspect of attack scalability. However, the underlying factors influencing transferability across diverse architectures are not yet well understood. Inconsistencies in transferability rates—particularly across

different types of models, such as convolutional neural networks and transformers—highlight the need for more research into the elements that govern cross-model success. Enhancing our understanding of these factors is essential for improving the consistency and reliability of adversarial attacks.

In summary, and in the context of the AIAS project, the following technology gaps will be addressed:

- Limited real-world applicability:
  - In AIAS, we will try to apply adversarial attacks against real-world AI models.
- Lack of understanding of black-box models:
  - In AIAS, we will try to develop and attack a custom black-box model.
- Adversarial attack automation:
  - In AIAS, we will try to automate the generation of adversarial AI attacks.

2.4. Adversarial AI attack mitigation

The field of adversarial AI attack mitigation is one that is undergoing rapid evolution, with a range of strategies having been developed with the aim of enhancing the resilience of AI models against malicious inputs designed to degrade system performance. The current range of mitigation approaches can be broadly categorised into three primary techniques: Data modification, model-based modifications and auxiliary defensive tools.

The objective of *data modification techniques* [BSS] is to alter the training data or input data in order to reduce vulnerability. To illustrate, adversarial training incorporates adversarial examples into the training set with the objective of enhancing robustness, whereas gradient hiding seeks to obscure model gradients in order to render it more challenging for adversaries to optimise attacks [QSR]. Other approaches, such as blocking transferability and data compression, seek to restrict the efficacy of adversarial examples that may impact disparate models or rely on reducing noise [HHB]. These techniques form a foundational layer of defence, rendering direct attacks on the model less effective. However, they are often constrained in their adaptability to evolving adversarial techniques.

Model-Based Modifications entail architectural alterations to the neural network itself, with the objective of enhancing security [SSG]. Such techniques include regularisation, which serves to reduce overfitting, and defensive distillation, which smooths model outputs in order to resist small perturbations. Another model-based technique, feature squeezing, involves the removal of superfluous input data details to mitigate the effects of adversarial attacks [HKD]. While these methods enhance model resilience, they entail trade-offs, such as increased computational demands and potential declines in benign performance. This may restrict their implementation in environments with limited resources, such as those typical of SMEs.

Auxiliary defensive tools serve to provide additional layers of identification and filtration of adversarial inputs during the process of model inference [CAA]. Defence-Generative Adversarial Network (GAN) and high-level representation guided denoisers represent examples of such tools [TKA], which employ generative and denoising techniques to detect adversarial noise. Such tools introduce a post-processing layer that facilitates the filtering of adversarial inputs during the inference phase, thereby establishing an additional defensive measure. The CALDERA platform [CAL], for instance, enables the emulation of adversarial behaviour, thus facilitating the testing and evaluation of security controls by cybersecurity teams. Similarly, the Atomic Red Team framework [ARO] automates the testing of general cyber defences. However, these tools are primarily optimised for conventional cybersecurity threats, which constrains their capacity to defend against the distinctive dynamics of adversarial AI.

Despite these advancements, the state of the art in adversarial AI attack mitigation remains constrained by an emphasis on general cyber threats as opposed to AI-specific attacks. This highlights the necessity for further progress so as to address the specific requirements of adversarial AI defence.

2.4.1 Identification of Technological Gaps

Although existing adversarial AI mitigation techniques have afforded a certain degree of protection, a number of significant technological deficiencies persist, emphasising the necessity for further advancement and enhancement.

Limited Simulation Platforms for Adversarial AI Attacks: The existing platforms, such as CALDERA and Atomic Red Team, facilitate the emulation of general cyber threats; however, they lack the capability to simulate adversarial AI-specific attacks in a realistic and tailored manner. This gap in capability limits the ability of cybersecurity teams to evaluate their defences against the nuanced and complex attack strategies targeting AI models. As a result, organisations are unable to assess vulnerabilities under conditions representative of real-world adversarial AI scenarios.

Lack of Comprehensive AI-Specific Mitigation Strategies: At present, standardised cybersecurity frameworks such as MITRE ATT&CK [MAT], Center for Internet Security [CIS], and Secure Controls Framework [SCF] offer mitigation strategies that are primarily focused on traditional cyber threats. These frameworks have yet to adapt to the specialised requirements of adversarial AI, which involve unique vectors such as evasion and poisoning attacks that specifically target machine learning models. In the absence of AI-specific guidance within these frameworks, organisations encounter difficulties in implementing targeted defences that address the particular challenges posed by adversarial AI.

Absence of Automated Adversarial AI Defence Testing: Automation frameworks that facilitate the rapid testing and validation of cyber defences are widely available in the field of traditional cybersecurity. Nevertheless, there is a notable deficiency in automated testing tools for adversarial AI defences. This deficiency in available tools leaves organisations reliant on manual or semi-automated approaches, resulting in a reactive rather than proactive defence posture and reducing the capacity for timely responses to evolving adversarial threats.

Insufficient Categorization of Mitigation Techniques for AI-Specific Attacks: A coherent and systematic classification of techniques for mitigating adversarial AI is currently unavailable. The absence of such a classification makes it challenging for organisations to identify the most appropriate techniques for different types of attacks, such as data poisoning versus evasion attacks. This deficiency hinders a strategic approach to adversarial AI mitigation, reducing the effectiveness of defence planning and deployment.

Gap in Comprehensive Evaluation Tools: The current suite of tools designed to assess the efficacy of adversarial AI defences suffers from two significant shortcomings. Firstly, many of these tools are overly generalised, focusing on traditional cyber threats rather than the specific challenges posed by AI systems. Secondly, they often lack the necessary comprehensiveness to effectively evaluate the layered approach required to safeguard AI systems. The absence of robust, adversarial AI-specific evaluation tools hinders the benchmarking of mitigation efficacy, leaving organisations without reliable measures to assess and enhance their defences.

2.4.2 Addressing Technological Gaps in Adversarial AI Mitigation in AIAS

The AIAS platform has been meticulously devised to address each of the identified deficiencies in the field of adversarial AI mitigation. It offers a comprehensive, adaptable, and scalable solution that is tailored to the specific needs of SMEs that are confronted with adversarial AI threats. The integrated architecture and

modular components of AIAS directly address the limitations of current adversarial defence technology, offering innovative approaches to simulation, testing and mitigation specifically for AI-based systems.

In order to address the limited availability of ***simulation platforms for adversarial AI attacks***, AIAS introduces a sophisticated ***Adversarial AI Engine*** that generates bespoke adversarial attack scenarios. In contrast to general cyber simulation platforms, this engine employs deep neural networks, such as GANs, and attack graph methodologies to generate attack vectors that closely resemble real-world adversarial tactics. The integration of these simulations into the platform allows organisations to test the robustness of their AI models under diverse adversarial conditions in advance, thereby enabling them to anticipate and prepare for AI-specific threats in a realistic and relevant manner.

In response to the ***absence of comprehensive AI-specific mitigation strategies***, AIAS incorporates a ***deception layer*** and a range of AI-driven detection and mitigation techniques that are explicitly focused on adversarial AI. This layer incorporates high-interaction honeypots, digital twins, and virtual personas that have been specifically configured to engage with adversarial AI attacks. By isolating and analysing these interactions, AIAS is able to capture valuable intelligence on adversarial methods, which is then used to inform the development of bespoke mitigation strategies. Moreover, the platform's mitigation framework incorporates recommendations based on XAI, which not only suggest specific actions but also provide transparency regarding these actions, thus assisting SMEs in understanding and executing optimal mitigation strategies that are aligned with the unique requirements of adversarial AI threats.

The ***lack of automated adversarial AI defence testing*** is addressed within AIAS through the ***AI-based Detection Module (AIDM) and LifeLong Reinforcement Learning (LLRL)***. These components guarantee continuous, automated monitoring and adaptation, thus enabling AIAS to respond proactively to new and evolving attack patterns without the necessity for manual intervention. By employing lLLRL, the detection module is capable of dynamically updating its threat detection algorithms in response to new data from both simulated and real attacks, thereby continuously enhancing its accuracy and response time. This automation effectively provides a proactive defence mechanism that evolves in real-time, thereby enabling SMEs to maintain a high level of protection even as adversarial tactics evolve.

To address the ***inadequate categorisation of mitigation techniques for AI-specific attacks***, AIAS has developed a ***comprehensive taxonomy of adversarial AI attacks*** within the Adversarial AI Engine. This taxonomy categorises adversarial attacks based on multiple dimensions, including the type of model targeted, the phase of the AI lifecycle under attack (training vs. inference), and specific attack vectors used (e.g., poisoning, evasion). This categorisation facilitates the development of bespoke mitigation responses, enabling AIAS to dynamically select and deploy the most appropriate techniques for each type of adversarial threat. The creation of a structured taxonomy by AIAS provides SMEs with a roadmap for the deployment of targeted defences, thereby facilitating the strategic application of mitigation techniques.

Finally, AIAS addresses the ***deficiency in comprehensive evaluation tools*** through its ***Security Data Fusion and Decentralised Knowledge Base***. These components aggregate and analyse security data from a variety of sources, including real-time attack interactions and simulated adversarial scenarios, in order to assess the effectiveness of AIAS's mitigation techniques on an ongoing basis. The utilisation of federated storage (i.e.,

through InterPlanetary File System (IPFS) and Hyperledger Fabric (a blockchain network)) enables the knowledge base to facilitate secure, decentralised data sharing across AIAS instances. This allows organisations to benchmark their defences against those of other SMEs and improve threat intelligence sharing. By centralising this data and applying sophisticated analytics, AIAS provides SMEs with a robust framework for evaluating and enhancing their adversarial AI defences based on continuous performance metrics and cross-organizational insights.

2.5. Security data fusion

Data gathering is an essential procedure for every AI-based task and several methods exist in the literature to collect data from a single or multiple sources. Web crawling and web scraping are two well-known methods that have been heavily deployed to create both small- and large-scale datasets.

Data managing systems become increasingly popular due to the advent of data lakes that facilitate individuals manage and process various datasets saving time and processing power. The benefits of combining data lakes with blockchain have been recently explored; however, little research can be found in the matter (as elaborated later).

AIAS will advance the state of the art by investigating and implementing a security data fusion approach that will combine data originating from different sources (e.g., detected cyberattacks, AI-systems’ vulnerabilities, and adversarial AI attacks) and data types (e.g., log files, network traffic). The security data fusion intents to combine IPFS with Hyperledger Fabric to create federated data storage. Moreover, AIAS also envisages to design and develop an AI-based web crawler to automatically collect data from the web including also dark web regarding adversarial AI attacks and cyberattacks, vulnerabilities in AI systems as well as malevolent information about the organization.

In order to do so, AIAS team has defined for the Data Fusion Module a systematic literature review that has been tackled based on a conceptual map as follows:

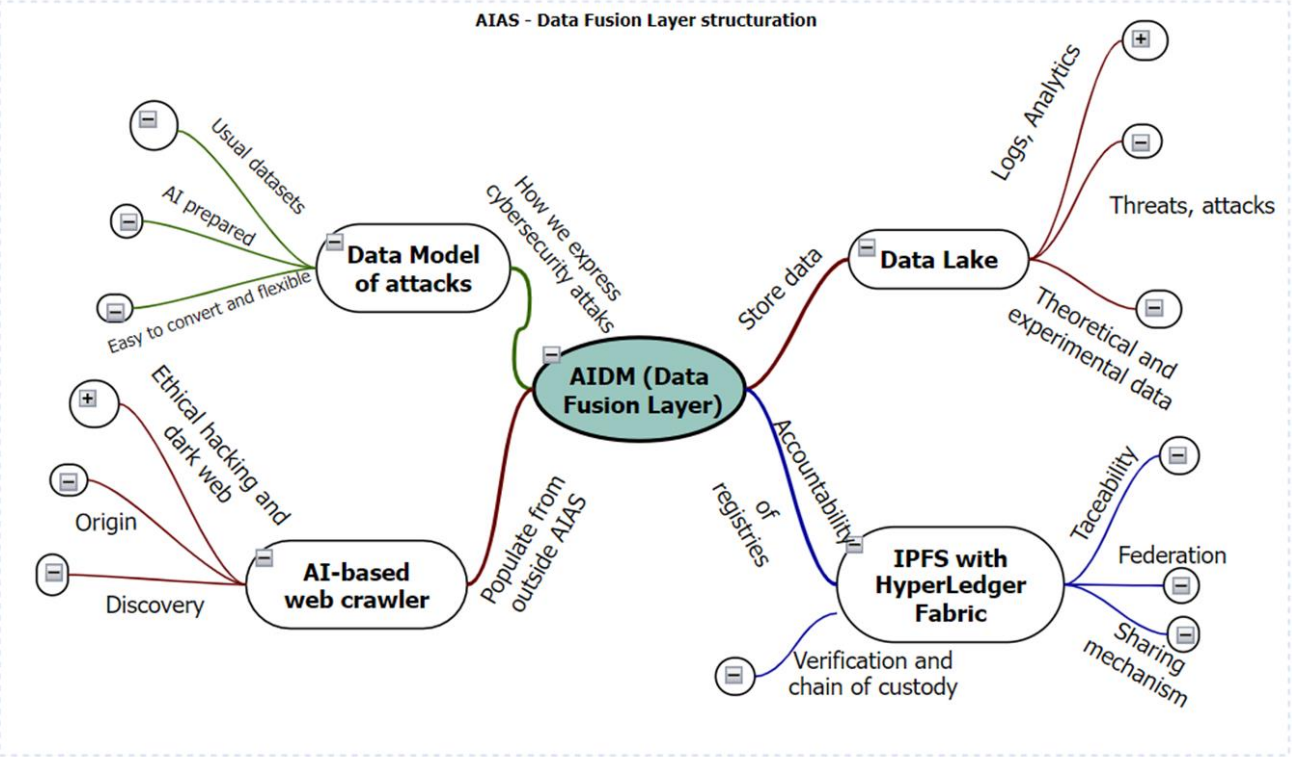

Fig. 1 Conceptual map of research lines and gaps for Data Fusion Module

The mentioned structuration, and the subsequent literature research performed has yielded the next reflections:

Security cyber-attacks data model from different sources

Due to the dynamic nature of cyber-threats, and the continued evolution of cyber-attacks, it is a very difficult task to create proper data models that properly represent in a simple yet accurate way the different cybersecurity events/issues that are captured in a system [AMN].

This section aims to provide an overview of the available material for cataloguing the attacks and the data sources that can be analyzed, with the intention of creating a baseline for AIAS’ AIDM module and to identify the needed innovations.

First, it is relevant to mention that several de-facto standards exist for cyber-attacks cataloguing. The most relevant is MITRE ATT&CK, an available database owned by the MITRE Corporation that keeps updated a long registry of type of cyberattacks. Those are retrievable from their exposed services in a specific format that will serve as inspiration for AIAS developments. Other relevant sources of information (considered as well standards for the community) are (Structured Threat Information Expression) STIX, (Common Attack Pattern Enumeration and Classification) CAPEC, (Common Vulnerabilities and Exposures) CVE or (Common Weakness Enumeration) CWE. A brief comparison among them is presented below (Table 5).

Table 5 De-facto standard cyber-attacks representation and modelling attempts

Standard	Format	Short reference
MITRE ATT&CK	JSON	Types and behaviour of adversaries for defence planning
STIX	JSON/XML	Representation of cyber-threats in machine-readable format
CAPEC	Web/schema	Taxonomy of attack patterns
CVE	Text	Catalog of vulnerabilities
CWE	Web/schema	Taxonomy of software weaknesses

Remarkably, STIX will also be explored by AIAS’ team, since the goal of machine-readable format of expressed attacks is of interest within the scope of the AI-Based Data Fusion Module for attacks detection.

Analyzing the available scientific literature, the most observable trend nowadays is the application of ML techniques over cyber-attacks (either expressed/available in the abovementioned standards or not). Prasad and Chandra [PRA] published in 2023 an in-depth review of the application of ML techniques for detecting and mitigating cyberattacks. There, the most important challenges were brough up (i.e., data imbalance, real-time information, variability, generalization or feature selection) as well as some future directions were given (better formatting and better datasets, transfer learning or the application of hybrid models), which assists AIAS’ team to identify the gaps to focus in during the project.

In this regard, some studies can be highlighted. The work [AMN] made an attempt to classify some cyber-attack modelling techniques. Those consist of mathematical or flow-based methods to characterize the different existing cyber-threats. The classification established: (i) Graph-based models, (ii) Petri nets, (iii) game theory models, (iv) Markov models and (v) ML/AI-based models. The work performed in such research might prove useful for the definition of data models for cyber-attacks (the expected activity in AIAS). While these models could theoretically be expressed in structured definition languages (e.g., JSON for attack trees or (Yet Another Markup Language) YAML for representing states in Petri Nets), the paper did not dig deep into the description and detail of types of cyber-attacks, which would be of interest to the goal of AIAS’ AIDM module.

Going beyond the aforementioned STIX standard, work [SHK] delved deep into a specific type of threat (APT, Advanced Persistence Threat), aiming to create datasets that adequately capture the complexity and subtlety of these attacks, which is a relevant perspective for AIAS (although the project is focused on adversarial attacks and not APTs). Also, in terms of modelling attacks, [SHK] elaborates on the MITRE ATT&ACK Framework, kill chain models and graph models as well.

Besides, paper [YCL] applied XGBoost, RNN (Recurrent Neural Network) and DNN (Deep Neural Network) models to detect cyberattacks over an Elasticsearch-Logstash-Kibana stack, focusing on network (NetFlow) logs [NFL]. A similar approach is visible in [SYJ], where NetFlow is also used to apply other types of ML models (distributed deep learning techniques) in a proposed data lake in order to boost further detection of cyberattacks.

From another angle, useful inspiration can be found in works such as [OKU], that aims at forecasting

cyberattacks from analyzing previous occurrences, historic cybersecurity datasets and/or synthetically generated attacks (relevant data sources, as per our interest definition). Although no real-time data was used, the paper rightfully addresses the challenges in the prediction, which mostly revolve around data incompleteness, imbalance, insignificance and noise. The paper proposed usage of ML models, and the application of advanced data handling techniques (e.g., balancing, feature selection, missing data) to improve the inference.

Concluding the reflection on the usage of ML for cyber-attacks modelling and detection, [TUS] reviews several techniques and frameworks used for detecting anomalies in cyber-physical systems, emphasizing the importance of selecting appropriate datasets for research in this field.

In the line of the above, an actionable lesson can be extracted from [OKU] and [TUS] such work is the different data sources used to perform the study:

- Usage of historic datasets of cyberattacks, including network intrusion attempts, failed login events, scanning activities, and other malicious behavior.
- Network traffic data, including packet sniffing, traffic volume, labelled samples and other metadata.
- Logs from server and other devices, including network information, error messages, authentication details, etc.

The exploration of different data sources is crucial for AIAS’ AIDM, since there is the goal of integrating (data fusion) registries coming from varying sources. The study of the usual data sources existing in the literature will help identify: (i) The types of data to be considered and (ii) the gaps that can be covered.

The work [LSF] explores the debate between using real-world data and synthetic datasets in the field of cyberattacks analysis. Real-world datasets are often preferred because they provide actual traffic logs and behavior patterns. However, the challenge with real-world data is the difficulty in differentiating between normal and malicious behavior, which can lead to issues in labeling and evaluating Intrusion Detection System (IDS) performance.

Works such as [AHM] have recently re-visited the datasets, methods and challenges for cyber-attacks detection, highlighting that the most relevant data sources are network traffic packets, Windows event logs, syslogs and historic datasets published by renowned actors (MITRE, CICIDS, etc.). They agree that a lack of more specialized, labeled datasets are needed, especially for distributed environments.

With regards to the most common datasets used in the studies (when historic registries are available), it has been noted that there are certain initiatives that periodically generate valuable material. Starting with the conventional DARPA 1998 IDS Dataset [DIDS] and KDD Cup 1999 [KDDC] in the early 200s, studies moved to use later NSL-KDD 2009 Dataset [KSLK]. With the explosion of worldwide access and use of the Internet, new datasets containing cybersecurity data were released, such as UNSW-NB15 2015 Dataset [UNSWN]. Also, CICIDS [CCIDS] has been a reliable source of cyber-attacks data via publishing a dataset every year in the period 2017-2020. More references can be found; nonetheless, a continuously updated collection is MAWI, made available by the WIDE (Widely Integrated Distributed Environment) Project in Japan, which broadcasts network traffic data captured from real-world Internet traffic in the form of daily *tcpdumps* files

(e.g., 2024 data in [MAW]).

However, neither the format, nor the coverage of the previous follow a common approach, and they lack a holistic view that can be integrated into a Data Fusion, multi-source environment such as in AIAS.

As mentioned, different system logs are one of the most productive data sources involved in the cybersecurity experiments found in the literature.

In 2021, [UHH] made an exhaustive analysis in more than 35 papers on the usage of logs as a relevant data source for cyber-attacks detection. Authors list the requirements of logs to be actionable; and propose a framework (called SOCBED [SOC]) that includes a log data generator that results of interest for AIAS, since it replicates real-world user behavior, system interactions and attack activities. Also, this work provides a flexible template through which some logs of a system can be generated out of configurations set up by users.

A more limited, yet still interesting, is found in [STE], where Windows event logs (from OS messages) are analyzed, modelled and replicated with the intention of detecting intruders. Since such logs are voluminous and complex, the work uses Natural Language Processing (NLP) techniques to sort them down and make them easier to manipulate. AIAS could benefit from the key-value pair formatting given to the logs’ entries, while the application of NLP techniques seem out of our scope. Also, the huge dataset used might be of interest for AIAS’ experimentation.

Authors in [ALJ] approach a similar issue, using ML models to drill down the logs of firewalls (which were pre-processed and pre-labelled as “normal”, “suspicious” or “malicious”) containing more than one million entries. Feature extraction suggested in the work might serve as an inspiration for AIAS’ data schemas for cyber-attacks detection.

One of the most interesting works is found very recently, where [KOK] attempted to devise a simple data schema for security logs identification; including “timestamp”, “event type”, “source IP”, “destination IP” and “severity” as the baseline attributes for further classification, accounting as well for anomalous entry identification. AIAS could leverage this material to (i) advance on the cyberattacks/log data schema definition and to (ii) synthetically generate logs whenever needed.

AIAS does not only aim at exploring the characterization of cyberattacks emanating from logs, but also from other non-conventional sources. In this regard, works like [IRS] explore the extraction of relevant cybersecurity intelligence information from unstructured textual reports such as incident reports, social media, news articles, and blog posts (e.g., Open-Source INTelligence (OSINT) [OSI]) and applying certain ML techniques (e.g., text mining, pattern recognition) to identify those.

As a conclusion in the gaps of literature, although there exist different de-facto standards for cyber-attacks and cyber-threats, there is not a current uncontested **taxonomy** or a **commonly formatted description** of attacks. Interestingly, the need for structured and interpretable attack representations is implied in various ways in the studied works. Worth noting, it is not the goal in AIAS to define such taxonomy, since comprehensive datasets and data schemas will always fail to fully emulate the multi-stage nature of certain cyberattacks. Only finding a compromise on a comfortable, actionable specification of such data format would be of application in our research action.

Datasets and data-source wise, traditional datasets often rely on data generated in isolated, controlled environments, such as virtual sandboxes, which do not capture the complexity and unpredictability of malware behavior in real-world conditions [RAD].

As a global observation by specialized researchers, there is a clear call for the development of **community-driven, open, anonymized datasets, so that further investigation (and in a more harmonized way) can be done to derive** where researchers can attack patterns, log generation scripts, or other tools; provided that reproducible datasets would exist if that situation is reached [LSF]. The current reality outlines that most log datasets used in publications are either the classic historic sets (as mentioned before: CICIDS, MITRE…) or data extracted by private organizations (e.g., [ALJ]), that are reluctant to share the raw information. However, notable exceptions exist such as [RAD], with 7 million network packets, 11.3 million OS system call traces and 3.3 million hardware event logs, or [DPD], a PCAP (Packet CAPture) file database of over one million instances that can be analyzed with packet sniffers such as Wireshark, and that was used for deep packet inspection in [DTP].

AI-based web crawlers

Web crawlers are systematic web crawling tools to collect information from the web pages found; they are based on the sequential tracking of links from one page to another according to pre-defined bases (e.g. page titles, websites URL, metatags and web page contents), thus building an index of the pages found according to their level of importance, which is based on aspects of the web page itself, such as: popularity, relevance and frequency of content update. Normally, web crawlers are used as search engines, but it is also very common to use them in conjunction with a scraper, which allows us to extract specific information from web pages. The web scraping process is divided into 3 stages [KHD]:

- **Fetching stage:** The desired website with the relevant information must first be accessed via the HTTP protocol, libraries such as *curl* and *wget* can be used by sending an HTTP GET request to the target address (URL) and get the HTML page as a response.
- **Extraction stage:** After retrieving the HTML page, the important data should be extracted. HTML parsing libraries, and XPath queries are utilized in this step; the XML Path Language (XPath) is a tool for finding information in documents.
- **Transformation stage:** Now that just the relevant data remains, it may be converted into a structured format for presentation or storage.

Continuing with the idea of web scrappers, the work [GEO] is based on generating information about potentially malicious hacking activities, using web scrapers and ontologies, which help to interpret human language and thus establish patterns of attacks and hacker behaviors. Therefore, the use of both provides us with an excellent tool for the extraction and collection of specific data.

Over the years, the use of web crawlers has been evaluated and sectorized, entering areas where data and statistics are essential, such as e-commerce, news and media analysis, public opinion control and of course

around research. Especially on cybersecurity and in line with AIAS, due to the lack of a centralized platform and efficient mechanisms for searching and collecting information on cybersecurity issues, in 2021, a tool was developed that proposed a mechanism to register, locate and share cybersecurity information using flexible structures based on XML and Resource Description Framework (RDF). It promised to improve access to critical information, minimize risks associated with outdated data, and facilitate collaboration with international entities on key cybersecurity issues [TAK].

Based on web crawlers, [WAN] demonstrates its use for vulnerability detection in software applications. The study proposes two forms of vulnerability detection, static detection, in which the source code is evaluated, and dynamic detection, which involves the analysis of the environment where the code is executed. The conclusions obtained were that web crawlers are a great tool for detecting and preventing common vulnerabilities such as SQL injections, Cross Site Scripting (XSS) and buffer overflows, however, for all this to be true and to be able to provide accuracy and reliability, manual inspection of the source code is essential.

However, all web crawler and web scraper systems present limitations and difficulties [BPO], which directly affect their operation and main objective, which is the collection and extraction of information, for example:

- **User Agent Identification**: Some crawlers identify themselves through user agent fields, which makes it easier for servers to block them.
- **Anti-Scraping Measures:** Many websites implement defense mechanisms such as locks and CAPTCHAs that make automated access difficult.
- **Dynamic Web Site Management:** Sites that use JavaScript and (Asynchronous JavaScript and XML) AJAX are difficult to crawl, as data is loaded dynamically.

Thus, knowing the problems presented by traditional web crawlers and web scrapers in terms of optimization and defense mechanisms, the need arises to automate the work and provide greater reliability and robustness to the models. Therefore, the next line of research in which AIAS will dig deep is to use AI to improve the initial performance of web crawlers and web scrapers and thus obtain AI-based web crawlers and AI-based web scrapers. Regarding AI-based web crawlers, [IBC] proposes a tool to reduce the workload of specialists through an automation approach that uses ML and NLD techniques, where the structure is based on 3 stages: (i) **Data collection, (ii) construction of prediction models, and (iii) data validation**; regarding AI-based web scrapers, [WEE] seeks to improve the efficiency and effectiveness of data extraction, as well as adaptability to dynamic sites.

Following this idea, valuable research has already been carried out on the architecture and evaluation of these models [KCT][KIM], where most studies show that the fundamental model of AI-based web crawlers is structured in 2 stages:

- **Crawling module:** Data collection through various sources.
- **Content classification:** Discrimination techniques to decide the nature of the analyzed data.

Another important research on AI-based web crawlers is collected in [KRI], where an AI-based web crawler system, *KnowCrawler*, focused on an AI-driven cloud architecture that processes data in parallel to optimize

classification and the use of ontologies to improve the relevance of the data, was designed and implemented. The goal of *KnowCrawler*, was to attack the problems discussed above about traditional web crawlers, basing its architecture in 3 parts: 1-Data Collection and Enrichment, 2-AI Based Classification and 3-Optimization and Priority setting; where the use of specific techniques in each module, such as WordNet 3. 0 for information processing (Module 1), the Bagging Classifier based on decision trees and random forest (Module 2), and Cuckoo Search for optimizing the selection of the most relevant URLs (Module 3), allowed obtaining very favorable results. The model was evaluated in terms of accuracy, retrieval, Harvest Rate and processing time, where we obtained metrics of Accuracy: 86.43%, Retrieval: 90.42%, Harvest Rate: 94.41% and Processing Time: 4.34ms (significantly lower compared to other models). Therefore, we can say that the KnowCrawler system represents a good reference in web crawling technology, using a cloud-based approach and ontologies can transform web crawling into a more powerful and scalable tool.
On the other hand, although the Internet is a huge structure that covers great aspects and which is in constant evolution; even so, most users can only access 4% of its entire extension, since the remaining 96% corresponds to classified and compromising information, in which the vast majority is strictly linked to illegal actions, such as weapon trading, child abuse, drug trafficking, etc.; this part of the Internet, is what is known as the Dark Web [ASH].

All the content of the Dark Web is hidden and not indexed, so the only way to access the Dark Web is through tools such as The Onion Routing (TOR) or the Invisible Internet Project (I2P), where, however, many pages have security mechanisms such as permissions and passwords for access [LPF].

The programs used to access the Dark Web provide the privacy of the data source as well as the privacy of the people who access the target data. The TOR technology/tool consists of retransmitting information through an immense network of nodes, in which each node encrypts the data, thus providing a high degree of privacy and anonymity. Each TOR user within the network has a random virtual circuit through which the data travels through the TOR nodes. After approximately 10 minutes, this virtual circuit changes, which makes it very difficult to identify the route, as well as the people involved, and the information transmitted. Thus, the TOR network has been developed as the most popular Deep Web technology.

During the last few years, the Dark Web has been widely studied, presenting great opportunities for the detection of possible cyber-attacks due to its nature of use for illicit actions [EPI][SCF][ARN][SCH]. For example, [EPI] in 2014 explored how signals and communication between actors on the dark web can provide information about future cyber threats. Complementarily, [SCF] implemented an effective monitoring system in the dark web, capable of deciphering and predicting emerging threats, thus helping cybersecurity analysts to proactively respond to risks.

Following the various studies and work carried out, the sharing of the process of extracting information about the Dark Web largely follows the following structure:

- **Collection:** Using Tor, the system anonymously accesses forums and collects raw data using automated browsers.
- **Processing:** Conversion of the data, for future use.
- **Analysis**: Data recognition and processing section.

- **Visualization:** Processed information is presented visually, allowing analysts to explore the data and detect trends in cybersecurity.

Real-time threat identification and tracking work has also been developed, such as [WSP], which addresses the growing need for proactive Cyber Threat Intelligence (CTI) (i.e., anticipation of potential cyber threat events) by incremental data collection and analysis (incremental analysis consists of querying, analyzing and extracting only new or updated data since the last collection, which increases efficiency by processing only the most recent information) in hacker forums, thus establishing an automated data classification through LSTM (Long Short-Term Memory) and maintaining a robust system capable of overcoming Anti-Tracking measures such as CAPTCHAs. The objective of the study was to improve the collection and analysis of threat intelligence in hacker forums. Building a system capable of identifying emerging trends and key actors in real-time, to improve threat response capability; for identification. In the process of data collection and classification, 2,930 files were obtained from 10 forums, where 59.32% belong to system exploits, 31.06% to network exploits and 9.6% to combined exploit threats. Thus, concluding with the obtaining of a system capable of identifying exploits, following a tracking model in web forums in real-time.

Some works [SAP][VLA] also deal with the development and implementation of systems and modules which use various data sources for the collection of possible information pertaining to cyber-attacks. In [SAP], authors developed DISCOVER, a system designed to generate early warnings of cyber threats by analyzing online conversations in social networks, cybersecurity blogs and dark web forums, seeking to identify words and terms that indicate possible cyber-attacks before they occur, helping to mitigate their impact. Conversation monitoring is conducted on: Twitter, Cybersecurity Blogs and Dark Web Forums. The alert system provides information such as the time in which the term was detected, the data source that generated it, the frequency of occurrence and the associated contextual words; alerts are generated if the word meets 2 criteria:

- **Frequency**: the term appears more than once in the time analyzed.
- **Context**: it matches words in the threat dictionary.

In terms of the results obtained, the DISCOVER model shows good results in terms of the accuracy of the alerts generated, with 81%, but if it is only based on blogs, the accuracy is 59%. Although this last value seems low, DISCOVER is an effective tool for early detection of cyber threats, where the system has already been tested, demonstrating its usefulness by identifying attacks such as Wannacry [WNC] and NotPetya [NTY] before they materialized.

As for work [VLA], a study was conducted for the detection of ongoing or imminent cyber-attacks using subtle signals from multiple unstructured sources. In this way, the SAINToS platform is developed, an innovative threat intelligence platform based on OSINT sources. It aims to complement traditional cybersecurity tools by integrating and analyzing data from social networks, the surface web and the dark web. SAINToS employs a modular architecture composed of four main subsystems, each responsible for collecting and analyzing data from different sources: Social Network Analyzer, Clear Net Crawler, Bug Bounties Analyzer and Deep Web Crawler. For each crawling module, patterns, actions and possible cyber-attack threats were identified, thus enhancing cyber-attack prediction and prevention capabilities through visual analysis and data correlation, making SAINToS a comprehensive cyber-attack collection and detection tool.

As cybercrime activities increase and become more organized, researchers consider the dark web a key environment for detecting and analyzing emerging cyber threats. Therefore, [BAS] explored recent studies in dark web content analysis, including methods, tools, techniques and results obtained. Thus, providing an overview of research focused on dark web analysis for threat detection. Most of the studies present the 4-phase architecture discussed above (Collection, Processing, Analysis and Visualization), focusing on the processes to overcome the security and privacy challenges presented by the dark web. The paper collects the analysis of 32 papers from 2017 to 2021, where summarizing the topics worked in each one, it is worth noting the little study and emphasis on the optimization of ML tools to carry out these tasks (of 32 studies only 4 deal with this part); being aware of this, from AIAS we will try to get into this sector of AI optimization.

In that way, knowing the limitations and difficulties presented by AI-based web crawlers and the dark web, comes into play this line of study and work by AIAS, which is to design an AI-based web crawler, in order to navigate, identify and collect information intelligently and efficiently, (including information from the Dark Web) using ML techniques to more accurately manage the collection of information, which will be processed and transferred to a database, which, together with the rest of the modules of the AIAS scheme, will form a data lake to store and manage all the information, following the general objective of the project, which is the detection of possible cyber threats and the development of a platform for the client to manage this problem.

Data lakes for combining info about cyber-attacks and threats

Data lakes have been in use as a way to centralize massive amounts of data for years, both structured and unstructured. They are a common tool used to run different types of analytics - from dashboards and data visualizations to big data processing and machine learning. Companies and organizations have been using data lakes to successfully generate business value from their data via real-time analytics, and in the cybersecurity sector, improving incident detection and response from the analysis of said data.

Data lakes allow for storing heterogeneous data types from multiple sources, including but not limited to: (i) Firewalls, (ii) IDS, (iii) network traffic, (iv) expected user behaviour. The data lake concepts that are key to cybersecurity are:

- **Centralized data storage:** The unification of all sources of data is the base of a data lake.
- **Data fusion**: The integration and comprehension of multiple different data sources, leveraging both historical data to recognize patterns of behaviour and real-time data to react in time to incoming threats. Some solutions go further beyond and introduce neural networks to classify data clusters.
- **Flexibility and structure:** Data lakes need to be able to adapt and be flexible to changes both in use and circumstances, which is in turn linked to its need to its structure - the schema of the data lake is not determined before the data is applied. Data is processed when it is being used.
- **Real-time data processing:** Many data lake solutions are integrated with real time analytic platforms, which goes hand in hand with the data fusion aspect of it to enable real-time monitoring and anomaly detection.

In the field of cybersecurity data lakes have multiple possible applications, some relevant fields found in literature are:

- **Real-time intrusion detection and threat analysis**. In literature we have examples of advanced heterogeneous data integration being used to collect and process data from sources such as firewalls and network logs that is then used by IDS as repositories of both historical and dynamic data. Frameworks are then used to prevent and detect multistage cyberattacks, significantly reducing false positives and increasing network-wide awareness.
  Data lakes allow for the storage of security data for prolonged periods of time, which in turn allows the cybersecurity teams to perform historical analysis. These analyses are valuable in incident response, allowing the teams to trace the evolution of the attack to learn from it.
  By using security data in this way Security Operations Centers (SOCs) can correlate data in real time. For example, if a distributed denial of service attack is underway or malware is present in the network, previous data can be used to detect the attacks before severe damage can be done. For these strategies, technologies such as machine learning are often used.
- **Anomaly detection and machine learning**. The usage of machine learning in data lakes has allowed the detection of unusual behaviours that are proof of cyberattacks or malware. For instance, machine learning algorithms can go through massive volumes of data to identify patterns of action and correlate them to attacks or intrusions if the network behaviour does not match what is expected or, on the contrary, matches prior detected attacks.
  Data lakes can be used to store and process structured, semi-structured and unstructured data, which is ideal for ML deployments. With machine learning theory working alongside data lakes, techniques such as clustering and regression analysis can be used to predict potential threats from behaviour patterns. Additionally, with new data being continuously provided they can adapt to evolving threats.
- **Forensic analysis**. The focus on data centralization of data lakes means that in the case of an attack all the available data is centralized in it, which means it is very useful in working for forensic analysis of attacks, as literature shows.
- **Threat intelligence and moving target defence**. One of the challenges of cybersecurity is keeping up to date with the attackers and the tools they have to perform cyber-attacks. Data lakes can be used here to gather data from feeds found across the internet to detect vulnerabilities and possible risks.
  This has the potential to match with Moving Target Defence (MTD), combining MTD with machine learning, threat intelligence and knowledge extracted from previous attacks. This way the strengths of MTDs can be leveraged to ensure the ever-changing environment remains one step ahead of the attackers.

Data lakes are not without problems hindering multiple challenges hinder in their effective integration and usage. One of the most agreed upon by literature is data quality and relevance. Since data lakes can store all manner of unstructured data, they often require cleaning and filtering processes in order to keep the data relevant and ensure its quality. This is even more crucial in cybersecurity since false positives are common

and can distract the security teams from the actual threats.

Another great challenge is the scalability of the data lakes, which can be a problem as large organizations collect ever increasing amounts of data, which can lead to performance issues. Data lakes must be optimized to ensure quick data retrieval, particularly for real-time threat detection.

Additionally, privacy and security concerns are increased even further for a data lake, since great quantities of potentially sensitive data may be stored in it. Ensuring proper encryption, access controls and monitoring is essential to prevent unauthorized access to the stored data.

IPFS with HyperLedger Fabric

The InterPlanetary File System (IPFS) is a peer-to-peer protocol and file sharing network for storing and sharing data in a distributed hash table. As opposed to a centrally located server, IPFS is built around a decentralized system of user-operators who hold a portion of the overall data. Any user in the network can serve a file by its content address, and other peers in the network can find and request that content from any node who has it using a distributed hash table.

The key features of IPFS are:

- Content addressing: Each file is split into smaller chunks, all hashed cryptographically and identified with their unique Content IDentifier (CID).
- Its distributed storage: All chunks are stored across a distributed network, ensuring redundancy.
- Data retrieval: Files must be retrieved from the nearest node hosting the required data.
- Versioning and immutability: It maintains a history of file versions, enabling audit trails and data restoration.

On the other hand, Hyperledger Fabric is an enterprise-grade permissioned distributed ledger framework for developing solutions and applications built by the Linux Foundation. It supports both private transactions and confidentiality via its permissioned open-source architecture. It uses smart contracts to specify the processes that must be executed at any given time. The code and the agreements contained therein exist across the distributed, decentralized blockchain network. Transactions are trackable and irreversible, creating trust between organizations and enabling businesses to make more informed decisions quicker—saving time and reducing costs and risks. The most relevant features of Hyperledger fabric are:

- Identity management: To enable permissioned networks, Hyperledger Fabric provides a membership identity service that manages user IDs and authenticates all participants on the network. (ii) Privacy and confidentiality: It allows for confidential data transactions, all data, including transaction, member and channel information, on a channel are invisible and inaccessible to any network members not explicitly granted access to that channel.
- Processing: transaction execution is separated from transaction ordering and commitment; this increases processing efficiency.
- Chaincode functionality: Chaincode applications encode logic that is invoked by specific types of transactions on the channel, ensuring that all transactions that transfer ownership are subject to the same rules and requirements.Modular design: all components can be customized and used separately.

Despite the advantages Hyperledger Fabric [VAF] has, it was never designed to handle large volumes of data due to scalability constraints and the high computational cost of storing massive amounts of data in the blockchain. Thus, the implementation with IPFS becomes possible and a valuable addition. Both technologies will be used in conjunction in the AIAS project as the base to build the AIDM, working one on top of the other to leverage the advantages of both while removing or minimising the disadvantages.

Some of the benefits of combining IPFS with blockchain technologies agreed upon by literature are:

- Efficient data storage: IPFS provides a cost-effective solution for storing large assets, while only the CID is stored in the blockchain.
- Increased performance: Leaving the large data assets to be handled by the IPFS reduces not only the size of the ledger, but also increases the performance of the system by reducing the computational costs associated with blockchain.
- Decentralized data sharing: The distributed nature of IPFS ensures that the data is still accessible even if some of the nodes go offline, while the blockchain maintains the status of the CIDs stored in it.
- Integrity and Trust: The nature of the blockchain ensures the stored data cannot be tampered or modified maliciously, it ensures that every action related to the handling of digital evidence is logged, making the evidence management process tamper-proof.

The integration of both technologies stands to combine the best of both technologies. Together they are a promising approach to addressing the challenges of data storage, security, and scalability in the AIAS project. Both technologies bring their strengths to the table, while mutually reducing the impact of their respective negative aspects.

3 Stakeholders

This section describes the key stakeholders and how they benefit from the AIAS platform.

Table 6 AIAS key stakeholder and their benefit

Stakeholder	Benefit
Organizations leveraging AI for their working operations	• Protection of their AI-based-systems from adversarial attacks, ensuring operational continuity and reducing potential business disruptions. • Proactive Detection and mitigation of threats. • Compliance with emerging security and ethical standards.
Cyber security professionals, managers and business consultants	• Access to cutting-edge adversarial AI defense and deception technologies that can be commercialized or integrated into existing products. • Advanced methods like generative AI, adversarial training and AI attack detection enrich their offerings. • Enhanced capabilities in both “AI for Cybersecurity” and “Cybersecurity for AI” improve their value proposition to clients.
Academia	• Promote commercialization of theoretical research and encourage European business to collaborate with academic institutions. • Create industry-academic alliances. • Cooperation among academic and industrial sectors with more expertise, which will help break down barriers between them. • Innovative research.
Civil Society	• Increased confidence in AI systems via enhanced protection against manipulation of AI-driven services ensuring reliability and trustworthiness to overall AI-based systems. • Reduced risk of data breaches and privacy violations due to robust adversarial defense mechanisms. • Ethical AI usage via transparent XAI solutions ensures ethical and fair decision-making processes that directly impact individuals.
Policymakers and Regulators	• Insights from the platform will help to shape regulations around AI and cybersecurity. • Identification of emerging threats in the AI landscape. • Adherence to ethical and legal standards via XAI mechanisms.

4 User and Technical Requirements

This section defines the user, functional and non-functional requirements of each AIAS component.

4.1. Methodology

AIAS partners have established a methodology through which the requirements are identified iteratively using an agile approach. A series of virtual meetings via MS Teams (including partners responsible for all project actions) are conducted to go through both user and technical requirements. After the proper discussions and conclusions, information is gathered using the whiteboarding technique to be later put in the appropriate templates (see Table 7, Table 8).

The methodology that has been used as a reference is Volere [VLR]. Volere has been used by thousands of organizations around the world to discover, define, communicate, and manage all the necessary requirements for any type of system development (e.g., software, hardware, commodities, services, organizational, etc.). Volere can be applied in all kinds of development environments, with any other development methods or with most requirements tools and modelling techniques. To produce accurate and unambiguous requirements, the Volere methodology uses techniques that are based on experience from worldwide business analysis projects and are continually improved.

The Volere methodology provides several templates to deal with the different techniques and activities that it includes. In a quick view, the Volere Requirement Process [VLR] suggests a methodology that has served as inspiration for the actions performed in AIAS:

- Analysis of project objectives and ambition
- Identification of the most crucial points where requirements can be extracted from
- Discussion of user requirements and documentation using a specific template
- Discussion of technical requirements and documentation using a specific template

Thus, the AIAS project consortium considered that choosing this methodology for T2.1 could help to achieve project objectives by structuring the gathered knowledge according to well established standards. Applying the Volere method for the requirement discovery process is essential to ensure that the real problem is addressed. The AIAS partners consider the method to be appropriate and to pave the way for development success. It is worth noting that similar approaches have been already employed by partners in previous successful projects (e.g., ASSIST-IoT, aerOS).

All requirements described in this document are identified during this first phase of the project (T2.1). As the project progresses the requirements may be polished, with potentially new ones to be included as they appear during the project so that they are continuously verified and, if necessary, adjusted.

The process of tackling AIAS user and technical requirements has been a result of studying exactly the meaning of those, and to ensure a proper transfer of the cited sequence of methodology into fully compliant documentation.

4.2. Definition of a requirement

While various definitions exist of what is a requirement, in this action, we agreed to use the definitions of ISO

and INCOSE:

“*A requirement is Statement that identifies a product (includes product, service, or enterprise) or process operational, functional, or design characteristic or constraint, which is unambiguous, testable or measurable, and necessary for product or process acceptability*.” (ISO/IEC 2007) [SSE]

“*A requirement is a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability*.” (INCOSE 2010) [INC]

Characteristics of requirements

The characteristics of good requirements are variously stated by different writers. There are several characteristics of requirements that are used to aid their development and verify their implementation into the solution (ISO 2011, Sections 5.2.5 and 5.2.6).

- ***Necessary:*** The requirement defines an essential capability, characteristic, constraint, and/or quality factor. If it is not included in the set of requirements, a deficiency in capability or characteristic will exist, which cannot be fulfilled by implementing other requirements.
- ***Appropriate:*** The specific intent and amount of detail of the requirement is appropriate to the level of the entity to which it refers (level of abstraction). This includes avoiding unnecessary constraints on the architecture or design to help ensure implementation independence to the extent possible.
- ***Unambiguous:*** The requirement is concisely stated. It expresses objective facts, not subjective opinions. It is subject to one and only one interpretation.
- ***Complete:*** The requirement sufficiently describes the necessary capability, characteristic, constraint, or quality factor to meet the entity need without needing other information to understand the requirement.
- ***Singular***: The requirement should state a single capability, characteristic, constraint, or quality factor.
- ***Feasible:*** The requirement can be realized within entity constraints (e.g., cost, schedule, technical, legal, or regulatory) with acceptable risk.
- ***Verifiable:*** The requirement is structured and worded in such a way that it will be possible to verify its accomplishment, as well as the degree of customer’s satisfaction regarding its realization.
- ***Correct:*** The requirement must be an accurate representation of the entity need from which it was transformed.
- ***Consistent:*** The requirement does not contradict any other requirement and is fully consistent with all authoritative external documentation.
- ***Comprehensible:*** The set of requirements must be written clearly to reflect what is expected by the entity and its relation to the system that it is a part of.

In addition, for those requirements that are related to technical components, it has been determined that AIAS will need to follow the SMART criteria [SMR]:

- ***Specific***. The requirements should precisely outline what the final product of AIAS needs. They should be clear, straightforward, consistent, and detailed enough to be understandable and actionable.

- ***Measurable***. Requirements need to be measurable. After the system is developed, it should be possible to confirm that each requirement has been fulfilled.
- ***Achievable***. Requirements should be practical and feasible within the project's limitations and the capabilities of the existing systems. Within AIAS, the participation of technical developers, system designers and cybersecurity experts will ensure that the specified requirements can be implemented.
- ***Relevant***. Requirements need to be directly related to the subject at hand and should align with the broader goals of the expected AIAS platform.
- ***Time-bound:*** A time-bound requirement specifies a clear deadline or timeframe within which the objective must be achieved, ensuring that progress can be tracked and deadlines met. This element provides urgency and helps prioritize tasks, making it crucial, as AIAS has a duration of 48 months.

The requirements’ type within the AIAS project are the following:

Once the definition and the characteristics of requirements were clarified, the ***following guiding principles*** it was established that requirements in AIAS can be of two **types:**

- *Technical requirements*: identify how the eventual product must fit into the world (i.e., the product might have to interface with or use some existing hardware, software, or business practice).
- *User requirements:* describe what the user needs and wants from the system, in terms of functionality, expected response, actuation capacity, visualization, etc.

The steps in the requirements capture procedure are the following:

The **methodology** used is a 5-step iterative process of identifying, capturing, defining, analysing, and reconciling the requirements (see Fig. 2). The requirements harmonization process steps are defined as follows:

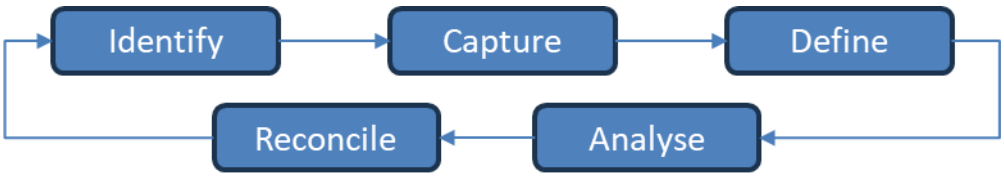

Fig. 2 The requirements capture procedure

- Identify sources of requirements

The first step is to identify new sources that can provide the know-how for requirements. In addition to our own knowledge, other sources could be stakeholders, regulation, standards, etc.

- Requirement Capturing

Make an inventory of identified requirements. This can be accomplished in a number of ways but in our case, each partner collects the requirements needed for their expected contributions, and then a joint discussion considering the global architecture platform is performed.

- Defining

Defining the information requirement is critical. Although the requirement is identified by the name,

completing the proposal requirement template is essential for the identification of the needs that the requirement explains.

- Analysing

A process of analysing the information is needed. It consists of assessing the requirements obtained. The following tasks need to be completed for each requirement:

  - Filling in the description
  - Correcting and homogenizing the relevant classifications
  - Grouping similar requirements
  - Validating requirements
  - Locating new requirements not identified in other sources of information

- Reconciling

This is the final step in which there is the agreement to incorporate the requirement into the list.

The AIAS requirements follow the prioritization as defined below:

In addition, a common requirement definition criterion adopted in AIAS was the **prioritization**. Project partners decided to follow the requirements prioritization proposed by the MoSCoW [MSC] methodology, that details as follow:

- ***Must have***: Requirements labelled as “MUST HAVE” have to be included as mandatory to be delivered in order for it to be a complete success. It is good to have clarity on this before a project begins, as this is the minimum scope for the product to be useful.
- ***Should have***: “SHOULD HAVE” requirements are also critical to the success of the project, but are not necessary for delivery in the final form. “SHOULD HAVE” requirements may be as important as “MUST HAVE”, although “SHOULD HAVE” requirements are often not as time-critical or there may be another way to satisfy the requirement.
- ***Could have:*** Requirements labelled as “COULD HAVE” are desirable but not necessary, and could improve user experience or customer satisfaction for little development cost.
- ***Won’t have***: Requirements labelled as “WON’T HAVE” have been agreed by stakeholders as the least critical.

For all this, two requirement templates have been prepared with the main information needed in order to be collected from the requirements identified. The Consortium decided to create a different template per each type of requirement. The rationale behind is to keep a natural separation, together with the fact that each of those might need different attributes (fields) to be completed in order to ensure a proper description. In the next paragraphs, the templates are exposed.

The user requirements template, as outlined in Table 7, represents the structured outcome of the AIAS methodology for gathering and refining requirements. This methodology, based on the iterative Agile principles, places emphasis on the systematic elicitation, documentation and prioritisation of user-centred needs. By translating user inputs into clear, actionable requirements, this template ensures alignment with AIAS's design and development goals, while providing traceability and modularity.

- **Requirement ID:** A unique identifier for each user requirement (e.g., USR-001). This ensures traceability and allows requirements to be referenced across all phases of the project lifecycle. During the ***Capturing and Documenting Requirements*** step, each user requirement is assigned an ID to maintain consistency and facilitate mapping to related technical requirements and dependencies.
- **Requirement:** A brief ***definition*** of the user requirement, emphasising the system's obligation to safeguard sensitive data during the training of AI/ML models (e.g., "AIAS MUST ensure the confidentiality and privacy of sensitive assets during the training of AI/ML models"). Relation to the Methodology: This field reflects the outcomes of the two preceding steps, namely the ***Identifying Sources of Requirements*** and ***Defining Requirements steps***. These steps entail the use of a variety of techniques, including stakeholder workshops, use-case analysis, and personas, with the aim of deriving precise and actionable user needs.
- **Priority:** The MoSCoW framework is employed for the categorisation of the importance of the requirement. (M: Must-have (Mandatory Requirement); S: Should-have (Desirable Requirement); C: Could-have (Optional Requirement); W: Will-not-have (Possible Future Enhancement)). The prioritisation is established during the analysis and prioritisation of requirements, wherein the feasibility, importance to the project's objectives and alignment with available resources of the requirements are evaluated.
- **Architecture Component:** This specifies the relevant architectural component that could address the requirement, which may be expressed as a name, such as "Security Data Fusion" or "Adversarial AI Engine." This is consistent with the ***Defining and Classifying Requirements*** phase, during which requirements are grouped according to their corresponding AIAS building blocks. The linking of user requirements to the relevant architectural components ensures that the system design is aligned with the identified user needs in a comprehensive manner.

Table 7 Template for user requirements

Requirement ID	**Requirement** (one line definition)	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
e.g. USR-001	e.g. (AIAS MUST ensure sensitive assets to remain private and confidential during the training of AI/ML models)	e.g. M	e.g. Security Data Fusion

The user requirements recording process is continuous throughout the lifetime of the project and technical requirements document is considered live. In practical terms, the collection of user requirements is done through an online spreadsheet that lives in the cloud document repository of the project.

The proposed methodology for gathering technical requirements (see Table 8) in AIAS is designed to integrate seamlessly with the Table 7. This integration offers a structured approach to translating elicited requirements into actionable and traceable components. The following section presents a detailed analysis of how these two frameworks complement each other.

Table 8 Template for technical requirements

	Template Field	Description
S	ID	Req-[Component]-[Type]-[ID]
	Dependencies	Dependences with user requirements
	Type	The classification of the requirements in Functional or non-functional and their specific Type that is linked directly to the relevant AIAS architecture component.
	Short name	A quick and self-explanatory name of the requirement
	Description	Description of the requirement, including info regarding its importance.
	Additional Information	Additional information relevant to the importance, unique characteristics and relevance of the requirement.
	Priority (MoSCoW)	Priority classification of the requirement.
M	Measures	Measurements and metrics regarding the implementation and validation of the requirement.
A	Achievable	Achievable within AIAS using the existing infrastructure and planned technologies (e.g., GANs for attack simulation, XAI for explainable recommendations).
R	Objectives	The tool MUST allow the CISO to carry out the risk assessment based on CNIL methodology, to enhance security.
T	Timeline	Year 1: independent test of functionality Year 2: verification of integration with other Application Building Blocks Year 3: validation against pilot use cases

1. **Traceability via Requirement ID:** The methodology places significant emphasis on the systematic identification and documentation of requirements, with each being assigned a unique identifier ***(Req-[Component]-[Type]-[ID])***. This approach aligns directly with the ID field in the template, thereby ensuring that every requirement is traceable throughout its lifecycle. By following the methodology's iterative refinement process, the ID field captures the component, functionality, and priority for accurate categorisation.
2. **Linkage of Dependencies:** The methodology encourages the reconciliation of user and technical requirements through the use of dependency mapping. This process is illustrated in the Dependencies field of the template. To illustrate, technical requirements pertaining to threat mitigation may be contingent

upon user requirements for real-time alerting. The iterative Agile approach guarantees that these dependencies are continually validated and updated.

3. **Requirement Classification (Types):** The methodology's step for defining and classifying requirements is reflected in the “type” field of the template. Requirements are classified as either functional (***FUNC***) or ***non-functional*** (***e.g. SECURITY, PERFORMANCE, USABILITY***) to guarantee that each technical requirement is explicitly aligned with its intended purpose and constraints. Additionally, the classification is expanded to reflect the specific types corresponding to AIAS's architectural building blocks, thus further enhancing traceability and modularity. By linking requirements to their respective AIAS components, the ***Type*** field categorizes them into the following:
   - **Adversarial AI:** For requirements associated with the Adversarial AI Engine, such as attack simulation and scenario generation.
   - **Deception:** For requirements tied to the Deception Layer, focusing on honeypots, digital twins, and virtual personas.
   - **Detection:** For requirements under the AI-Based Detection Module (AIDM), including real-time anomaly detection and continuous learning.
   - **Mitigation:** For requirements in the XAI-Based Mitigation Engine, such as explainable recommendations and mitigation strategies.
   - **DataFusion:** For requirements related to the Security Data Fusion Component, encompassing threat data aggregation and analysis.
   - **KnowledgeBase:** For requirements involving the Decentralized Knowledge Base, ensuring secure storage and access to threat intelligence.
   - **ThreatIntel**: For requirements specific to collaborative threat intelligence sharing and interoperability.
   - **AccessControl:** For requirements under the Authentication and Access Control Manager, ensuring secure user access and authorization.
   - **Scalability:** For requirements associated with the Scalability and Resource Management Module, ensuring resource optimization and system scalability.
4. **Clear and Actionable Descriptions**: The methodology's emphasis on the ***SMART*** criteria (specific, measurable, achievable, relevant, and time-bound) aligns with the Description field in the template. This ensures that each requirement is articulated with precision, detailing what the system must achieve and how it contributes to AIAS's objectives, such as the mitigation of adversarial AI.
5. **Additional Context**: During the requirements gathering phase, supplementary information is frequently collated to provide context or clarify constraints. This directly corresponds to the Additional Information field in the template, thus enabling the inclusion of supplementary details as required.
6. **Prioritisation and Feasibility**: The ***MoSCoW*** prioritisation framework, as outlined in the methodology, is integrated seamlessly with the Priority (MoSCoW) field in the template. This ensures that requirements are classified based on their necessity, for example, as a '***Must-have***' or a '***Should-have***'. Furthermore, the Achievable field in the template aligns with the methodology's iterative refinement step, whereby each requirement is validated in terms of its feasibility within AIAS's scope and technological boundaries.
7. **Validation Metrics:** The methodology's emphasis on continuous validation is reflected in the Measures

field of the template. Metrics for ***implementation*** and ***validation*** (e.g., detection accuracy or compliance rates) are essential for evaluating whether the requirements are achieving their intended outcomes.

8. **Achievability:** The Achievable field in the template is closely aligned with the methodology's emphasis on validating feasibility during the ***Analysing and Prioritizing Requirements step***. This ensures that each requirement is not only aligned with AIAS's overarching objectives, but also based on a realistic assessment of technological capabilities, resource availability and timeline constraints. To illustrate, a requirement may be considered achievable within AIAS if it utilises existing infrastructure and planned technologies, such as employing GANs for attack simulation or XAI frameworks for explainable recommendations. Nevertheless, its successful implementation may be contingent upon the availability of sufficient computational resources and the timely integration of foundational components, including the Adversarial AI Engine. This field serves as a pivotal checkpoint to guarantee that requirements are practical and actionable, striking a balance between ambition and the realities of AIAS's development environment.
9. **Objectives Alignment**: The methodology focuses on aligning requirements with ***AIAS’s overarching goals***, such as privacy-preservation and adversarial robustness. This is explicitly captured in the Objectives field of the template, which defines the purpose and relevance of each requirement in enhancing AIAS’s capabilities.
10. **Timeline Integration**: The methodology's iterative approach encompasses phased deliverables and milestone tracking, which is documented in the Timeline field of the template. To illustrate, the initial stages of testing are aligned with the preliminary assessment of standalone functionality, whereas subsequent phases concentrate on the integration and validation of the system in realistic operational scenarios.

The technical requirements recording process is as well continuous. In this regard, the collection of technical requirements is done through creating a single table per each one of the technical requirements (the requirement fiche), and an updated version of every fiche is kept in the cloud document repository of the project.

4.3. User, Functional & non- Functional requirements of AIAS modules

This Section includes the defined user, functional and non-functional requirements of each AIAS module.

4.3.1 AIAS Deception mechanism

A key part of the AIAS project is the deception module. This means we need to focus on how the deception mechanisms work, as they should be able to distract attackers and drain their resources, diverting them from the real systems while collecting valuable information. By meeting these goals, we enhance both the effectiveness and strength of the deception layer.

Table 9 Functional and non-functional requirements of the AIAS deception mechanism

		Description
S	ID	REQ-DECEPTION- DEC-1
	Dependencies	N/A

	Type	DEC: Decoy
	Short name	Flawless Imitation
	Description	The honeypot MUST respond to every request exactly as the attacker would expect from a real system. It needs to replicate all system functions, including generating accurate error messages. Communication protocols MUST appear completely authentic.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Validation of response accuracy through simulated attacks.
A	Achievable	Use existing honeypot frameworks and virtual personas.
R	Objectives	The honeypot should fully imitate a real system to effectively deceive attackers and gather intelligence.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION- DEC-2
	Dependencies	N/A
	Type	DEC: Decoy
	Short name	Controlled Honeypot Discrepancy
	Description	The honeypot MUST have slight differences from the real system to deceive attackers while safeguarding the true structure. The simulation can be detailed but should stop before revealing sensitive system elements. Less discrepancy may increase exposure.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Analysis of system discrepancies through controlled attacks and monitoring attacker behaviour.
A	Achievable	The discrepancies can be chosen by selectively omitting or altering non-essential features.
R	Objectives	The honeypot should mimic the real system with enough detail to deceive but maintaining a level of controlled inaccuracy to protect sensitive information.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION-SEC-3
	Dependencies	N/A
	Type	SEC: Security
	Short name	Secured Deception System Communication

	Description	The communication between components outside the deception system MUST be secured to prevent unauthorized access or data leaks.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Validation through network security tests.
A	Achievable	Integration of industry-standard encryption protocols, firewalls, and secure APIs.
R	Objectives	The communication channels outside the deception system should remain fully secure, preventing any exposure of sensitive data.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION- DEC-4
	Dependencies	N/A
	Type	DEC: Decoy
	Short name	Fake Cooperation
	Description	The honeypot SHOULD deceive the attacker by simulating the progression of their attack, only to cause a failure or crash at the final stage. This approach allows for better monitoring and analysis of the attacker’s behaviour throughout the process.
	Additional Information	None
	Priority (MoSCoW)	S: Should-have. Desirable Requirement
M	Measures	Test simulations showing how attackers react to fake progress.
A	Achievable	This is achievable by implementing staged responses that mimic the real system, leading attackers down a controlled path.
R	Objectives	Gather more information about the attacker’s methods and techniques by allowing them to believe their attack is succeeding.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION- DEC-5
	Dependencies	N/A
	Type	DEC: Decoy
	Short name	Attack Traceability through Honeytokens
	Description	Honeytokens COULD be used to track an attacker’s actions within the honeypot. If an attacker accesses fake data, such as credentials, and tries to use them elsewhere, their movements can be tracked, providing a clear trail of their activity.
	Additional Information	None

	Priority (MoSCoW)	C: Could-have. Optional Requirement
M	Measures	Monitor the usage of honeytokens to trace attacker movements.
A	Achievable	This can be implemented by placing honeytokens such as fake credentials or documents in key locations within the honeypot.
R	Objectives	To gather detailed information about an attacker's actions and strategies by tracking their interaction with fake data, leading to better analysis of attack patterns.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION- DEC-6
	Dependencies	N/A
	Type	DEC: Decoy
	Short name	Adversary Diversion Time
	Description	The honeypot COULD maximize the time an attacker spends interacting with the fake system. This can be achieved by mechanisms like simulating slow network connections, adding response delays, limiting connection numbers, or creating complex virtual environments with multiple open ports.
	Additional Information	None
	Priority (MoSCoW)	C: Could-have. Optional Requirement
M	Measures	Monitor the time attackers spend interacting with the honeypot. Test different methods (delays, limiting connections, complex topologies) to see which are most effective at extending interaction time.
A	Achievable	This is achievable by implementing time-delay mechanisms and resource draining tactics within the honeypot.
R	Objectives	To divert and delay attackers, forcing them to spend more time interacting with the honeypot, which reduces the time and resources they have to attack real systems.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION-DEC-7
	Dependencies	N/A
	Type	DEC: Decoy
	Short name	Attack Redirection
	Description	The honeypot MUST redirect malicious traffic away from the real system and towards the deceptive layer when adversaries are detected. This ensures that attackers interact only with the fake environment, protecting the real infrastructure.
	Additional Information	None

	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Monitor network traffic and validate redirection mechanisms through attack simulations.
A	Achievable	This can be achieved using networking tools and configuring routers or firewalls to automatically redirect malicious traffic to the honeypot.
R	Objectives	To ensure that all detected malicious activity is redirected to the deceptive environment, minimizing the risk to the real system, and allowing for safe attack analysis.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION-DEC-8
	Dependencies	N/A
	Type	DEC: Decoy
	Short name	Data SOULD appear Realistic, Protected, and Consistent in the Honeypot
	Description	The honeypot’s data must: a. Look Real: The content of the data should resemble real information. It should appear meaningful, even though it’s invalid. b. Look Protected: Although the data is fake, it should not be too easy for the attacker to access, as that might be suspicious, and information will seem worthless. c. Look Consistent: Changes made by the adversary should remain in the current and next sessions to maintain the illusion of a real system.
	Additional Information	None
	Priority (MoSCoW)	**S**: Should-have. Desirable Requirement
M	Measures	Test security layers to ensure attackers face realistic challenges. Validate consistency by tracking data persistence across multiple sessions.
A	Achievable	Achievable by generating realistic but invalid datasets, deploying security measures like authentication to simulate protection and state-tracking mechanisms to ensure data consistency between sessions.
R	Objectives	The honeypot should simulate realistic and protected data that behaves like a real system, while preserving the illusion of consistency.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION-NFR-9
	Dependencies	N/A
	Type	NFR: non-functional requirement
	Short name	Performance, Synchronization, and Interoperability

	Description	The honeypot or digital twin MUST ensure: a. Performance: The honeypot or digital twin must operate efficiently without significantly impacting the overall system’s resources during interactions with attackers. b. Synchronization: The honeypot or digital twin must reflect real-time changes in the actual system, maintaining the consistency. c. Interoperability: The communication between the digital twin and the target system must be continuous, and bidirectional, allowing data exchange and system interaction.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Performance will be measured through system resource monitoring during honeypot interactions. Synchronization and interoperability will be tested through real-time updates and communication between the honeypot and the actual system.
A	Achievable	Achievable by optimizing resource usage through efficient system design and leveraging real-time data synchronization protocols. Interoperability can be achieved using standard APIs.
R	Objectives	The honeypot should operate without affecting the performance of the main system, stay synchronized with real-time updates, and allow continuous communication between both environments.
T	Timeline	M20
		Description
S	ID	REQ-DECEPTION-NFR-10
	Dependencies	N/A
	Type	NFR: non-functional requirement
	Short name	Maintenance and Reliability
	Description	The system SHOULD ensure: a. Maintenance: The tool should easily accept updates, new configurations, and changes, with emphasis on supporting modular architectures. b. Reliability: The deception mechanism should be capable of functioning under any condition and be able to alert (and if possible, recover) from unexpected issues, ensuring consistent and reliable operation.
	Additional Information	None
	Priority (MoSCoW)	**S**: Should-have. Desirable Requirement
M	Measures	Test update mechanisms, configuration changes and modular integration. Validate reliability through stress tests and recovery scenarios.
A	Achievable	Achievable by designing the system with modular components and robust update procedures.

R	Objectives	To ensure that the tool can be easily updated and configured while maintaining high reliability and quick recovery from any operational issues.
T	Timeline	M20

The following user-centred requirements ensure that users have a seamless experience when interacting with the system.

Table 10 User requirements of the AIAS Deception mechanism

Requirement ID	**Requirement** (one line definition)	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
USR-001	AIAS MUST ensure sensitive assets to remain private and confidential during the training of AI/ML models	M	Security Data Fusion
USR-002	The system SHOULD provide clear feedback about its decisions and actions.	S	Deception Module
USR-003	Results MUST follow guidelines to be understandable across the environment and minimize ambiguity.	M	Monitor and Security Analytics
USR-004	Users SHOULD be able to adjust the level of autonomy or deception employed by the system.	S	Deception Module
USR-005	The system SHOULD allow users to provide feedback on false alarms and missed deceptions for improvement.	S	Deception Module
USR-006	Users will expect the system to accurately detect attacks with minimal error.	M	Deception Module

4.3.2 AIAS Detection mechanism

This section includes the requirements of the AIAS Detection mechanism.

Table 11 Functional and non-functional requirements of the AIAS Detection mechanism

	Template Field	Description
S	ID	Req-Detection- SEC-1
	Dependencies	USR-007
	Type	SEC: Security
	Short name	Real-Time Detection
	Description	The AIAS platform MUST detect adversarial inputs in real-time to prevent data manipulation and attacks.
	Additional Information	Requires continuous monitoring
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Implementation and validation.
A	Achievable	Achievable using AIAS detection algorithms.
R	Objectives	Enable proactive threat identification in real-time.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-FUNC-2
	Dependencies	USR-008
	Type	FUNC: Functional
	Short name	Customizable Detection Rules
	Description	Users MUST be able to configure detection rules based on specific operational environments and threat profiles.
	Additional Information	Requires flexible rule-set configuration
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Implementation and user testing
A	Achievable	Achievable through adaptive detection algorithms.
R	Objectives	Allow users to tailor detection mechanisms for specific threats.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-FUNC-3
	Dependencies	USR-009
	Type	FUNC: Functional

	Short name	Anomaly Logging
	Description	The platform SHOULD log all detected anomalies for detailed analysis and historical trend review.
	Additional Information	Integration with analytics tools
	Priority (MoSCoW)	S: Should-have. Desirable requirement.
M	Measures	Implementation and review
A	Achievable	Achievable through AIAS's logging modules.
R	Objectives	Provide comprehensive logs for forensic analysis.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-FUNC-4
	Dependencies	USR-010
	Type	FUNC: Functional
	Short name	Instant Notifications
	Description	Users SHOULD receive immediate alerts upon detection of adversarial inputs to enable swift action.
	Additional Information	Integration with alerting systems
	Priority (MoSCoW)	**S**: Should-have. Desirable Requirement
M	Measures	User feedback and validation
A	Achievable	Achievable using existing notification modules.
R	Objectives	Ensure timely responses to potential threats.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-SEC-5
	Dependencies	USR-011
	Type	SEC: Security
	Short name	Multi-Method Detection
	Description	The system MUST employ multiple techniques, including anomaly detection and pattern recognition.
	Additional Information	Utilizes AIAS detection modules
	Priority (MoSCoW)	M: Must-have. Desirable requirement.
M	Measures	Algorithm performance testing
A	Achievable	Achievable with AIAS's modular detection approach.

R	Objectives	Enhance detection accuracy through diverse methodologies.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-SEC-6
	Dependencies	USR-012
	Type	SEC: Security
	Short name	Minimize False Positives
	Description	The detection module MUST minimize false positives to avoid operational burden and unnecessary alerts.
	Additional Information	Requires machine learning optimization
	Priority (MoSCoW)	**M**: Must-have. Desirable Requirement
M	Measures	Accuracy and performance evaluation
A	Achievable	Achievable through model training and tuning.
R	Objectives	Improve the reliability of detection mechanisms.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-SEC-7
	Dependencies	USR-013
	Type	SEC: Security
	Short name	Adaptive Detection Parameters
	Description	The AIAS platform COULD automatically adapt detection parameters based on evolving attack patterns.
	Additional Information	Utilizes machine learning for adaptability
	Priority (MoSCoW)	**C**: Could-have. Optional Requirement
M	Measures	Continuous monitoring and adjustment
A	Achievable	Achievable with AIAS adaptive learning models.
R	Objectives	Ensure detection mechanisms evolve with new threats.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-FUNC-8
	Dependencies	USR-014
	Type	FUNC: Functional
	Short name	Historical Incident Database

	Description	Users SHOULD access a historical database of incidents for trend analysis and strategy development.
	Additional Information	Requires secure storage solutions
	Priority (MoSCoW)	S: Should-have. Desirable requirement.
M	Measures	Implementation and usability testing
A	Achievable	Achievable with AIAS's data storage modules.
R	Objectives	Provide insights into past incidents for better preparation.
T	Timeline	M30
	Template Field	Description
S	ID	Req-Detection-FUNC-9
	Dependencies	USR-015
	Type	FUNC: Functional
	Short name	Performance-Friendly Operation
	Description	The detection module SHOULD operate without impacting the performance of other critical system operations.
	Additional Information	Requires performance optimization
	Priority (MoSCoW)	**S**: Should-have. Desirable Requirement
M	Measures	Performance testing and validation
A	Achievable	Achievable through AIAS's efficient resource usage.
R	Objectives	Ensure the system’s normal operations remain unaffected.
T	Timeline	M30

Table 12 User requirements of the AIAS Detection mechanism

Requirement ID	**Requirement** (one line definition)	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
USR-007	The AIAS platform MUST detect adversarial inputs in real-time to prevent potential attacks and data manipulation.	M	Detection Mechanisms

USR-008	The system MUST allow users to customize the detection rules based on specific operational environments and threat profiles.	M	Detection Mechanisms
USR-009	The platform SHOULD provide detailed logs for all detected anomalies to facilitate further analysis by the security team.	S	Detection Mechanisms
USR-010	Users SHOULD receive instant notifications on their dashboard when a potential adversarial attack is detected.	S	Detection Mechanisms
USR-011	The AIAS system SHOULD utilize multiple detection techniques, including anomaly detection and pattern recognition, to enhance accuracy.	S	Detection Mechanisms
USR-012	The detection module SHOULD operate with minimal false positives to avoid unnecessary alerts and reduce operational burden.	S	Detection Mechanisms
USR-013	The AIAS platform COULD automatically adapt its detection parameters based on evolving attack patterns in the operational environment.	C	Detection Mechanisms
USR-014	Users SHOULD be able to review past detected incidents through an easy-to-navigate historical database for trend analysis.	S	Detection Mechanisms
USR-015	The detection module SHOULD operate seamlessly without impacting the performance of other critical system operations.	S	Detection Mechanisms

4.3.3 AIAS Adversarial AI (Weaponizer)

This section includes the requirements of the AIAS adversarial AI component.

Table 13 Functional and non-functional requirements of the AIAS adversarial AI component

		Description
S	ID	REQ-WEAPONIZER-FUNC-1

	Dependencies	N/A
	Type	Functional
	Short name	Taxonomy of adversarial AI attacks
	Description	The system MUST have available the taxonomy of adversarial AI attacks
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	At least 2 attacks.
A	Achievable	Achievable by studying the state of the art of adversarial AI attacks
R	Objectives	To ensure that the Weaponizer covers state-of-the-art attacks
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-02
	Dependencies	N/A
	Type	Functional
	Short name	Information collection
	Description	The system MUST collect information from target systems. This information is used to train the AI model and eventually exploited to perform adversarial AI attacks.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	At least 2 attacks.
A	Achievable	Achievable by designing internally the data collector.
R	Objectives	To ensure that the Weaponizer behaves correctly and has all the needed information.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-3
	Dependencies	N/A
	Type	Functional
	Short name	Attack Graphs
	Personas	
	Description	The system MUST implement attack graphs. In order to automatically generate adversarial AI attacks, the attack graph of a specific AI model must be available to the weaponizer.
	Additional Information	None

	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	At least 1 attack.
A	Achievable	Achievable by designing internally the target AI model.
R	Objectives	To ensure that the Weaponizer has all the needed information to perform adversarial AI attacks.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUN-4
	Dependencies	REQ-WEAPONIZER-FUNC-1, REQ-WEAPONIZER-FUNC-2, REQ-WEAPONIZER-FUNC-3
	Type	Functional
	Short name	Weaponizer feeding
	Description	The system MUST feed information collected by the work done to meet the technical requirements which are dependencies of this one. This data is required by Weaponizer to specifically generate and perform attacks on the target AI model.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Weaponizer completeness.
A	Achievable	Achievable by designing and implementing APIs between Weaponizer and its internal modules.
R	Objectives	To ensure that the Weaponizer has all the needed information to perform adversarial AI attacks.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-5
	Dependencies	N/A
	Type	Functional
	Short name	Required hardware
	Description	AIAS consortium will develop the Weaponizer on top of dedicated environment. Also, dedicated hardware is needed to train the target AI model, to collect the data, to execute attack graphs, and finally to evaluate the results.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	N/A
A	Achievable	Achievable by buying/renting/using existing dedicated hardware

R	Objectives	To ensure that the Weaponizer has enough computational power to be executed.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-6
	Dependencies	N/A
	Type	Functional
	Short name	Target AI model
	Description	Design and implement a target AI model to use as use-case for AIAS.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	The Weaponizer works as expected.
A	Achievable	Achievable by exploiting the transfer of knowledge of AIAS consortium.
R	Objectives	To ensure that the Weaponizer has a target AI model.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-7
	Dependencies	N/A
	Type	Functional
	Short name	Taxonomy
	Description	Create a taxonomy of adversarial AI attacks.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	The Weaponizer covers specific adversarial attacks.
A	Achievable	Achievable by studying state of the art and collecting existing adversarial AI attacks.
R	Objectives	To ensure that the Weaponizer has knowledge of existing adversarial attacks.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-8
	Dependencies	REQ-WEAPONIZER-FUNC-6
	Type	Functional
	Short name	Taxonomy
	Description	MUST test and implement adversarial AI attacks on the target AI model.
	Additional Information	None

	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	The Weaponizer must attack an AI model.
A	Achievable	Achievable by implementing internally an AI model for the specific use-case.
R	Objectives	To ensure that the Weaponizer has attack capabilities on AI models.
T	Timeline	M30
		Description
S	ID	REQ-WEAPONIZER-FUNC-9
	Dependencies	N/A
	Type	Functional
	Short name	Weaponizer API
	Description	Define an API between the attack graph and the adversarial attack generator.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	The Weaponizer modules must be connected to each other.
A	Achievable	Achievable by defining and implementing an API format.
R	Objectives	To ensure that the Weaponizer modules can transfer information among each other.
T	Timeline	M30

Table 14 User requirements of the AIAS adversarial AI component

Requirement ID	**Requirement** (one line definition)	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
USR-016	Users should be able to search and analyse Weaponizer logs, to identify reasons and TTPs of the adversarial AI attacks	S	Weaponizer

4.3.4 AIAS Mitigation mechanism

The following user-related requirements are closely aligned with the mitigation-related functionalities of the AIAS framework. They emphasise real-time responses, human-in-the-loop approaches and explainability,

while ensuring integration with other components, such as the Adversarial AI Engine and the Security Data Fusion.

Table 15 User requirements of the AIAS mitigation mechanism

Requirement ID	Requirement	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
USR-017	The AIAS Mitigation Engine MUST provide real-time explainable recommendations for countering adversarial AI attacks.	M	XAI-Based Mitigation Engine
USR-018	The AIAS platform SHOULD allow human operators to modify and approve mitigation actions suggested by the Mitigation Engine.	S	XAI-Based Mitigation Engine
USR-019	The Mitigation Engine MUST ensure the effectiveness of proposed actions by analysing historical data and past attack scenarios.	M	Security Data Fusion
USR-020	The AIAS Mitigation Engine MUST support integration with the Adversarial AI Engine to dynamically learn from new attack scenarios.	M	Adversarial AI Engine
USR-021	The Mitigation Engine SHOULD provide probabilistic scoring of mitigation strategies to guide human decision-making.	S	XAI-Based Mitigation Engine

The user requirements for the AIAS Mitigation Engine are classified as either "MUST" or "SHOULD" based on their criticality to the engine's core functionality and their alignment with AIAS's objectives. It is imperative that the requirements classified as MUST, such as providing real-time explainable recommendations (USR-017), ensuring effectiveness through historical data analysis (USR-019), and dynamically learning from new attack scenarios (USR-020), are met for the engine to fulfil its primary role of mitigating adversarial threats.

The provision of real-time recommendations is fundamental to the minimisation of the impact of attacks, while the assurance of explainability is essential to the fostering of operator trust and comprehension of the proposed actions. Similarly, the utilisation of historical data is of paramount importance for the enhancement of mitigation strategies and the avoidance of repeated failures, thus making it an indispensable component of reliable decision-making. Furthermore, dynamic learning through integration with the Adversarial AI Engine is also imperative, as this enables the engine to adapt to evolving threats and maintain relevance in a rapidly changing threat landscape.

In contrast, the SHOULD requirements, such as enabling human operators to modify and approve actions (USR-018) and providing probabilistic scoring of strategies (USR-021), are highly desirable but not essential for the engine's core functionality. These features enhance the usability of the system and operator trust by supporting human-in-the-loop operations, which are particularly valuable in high-stakes scenarios. Furthermore, probabilistic scoring facilitates decision-making by providing a quantitative assessment of potential mitigation strategies, which aligns with the engine's role as a decision support system. In combination, these requirements establish a robust framework for the Mitigation Engine, striking a balance between essential functionalities and enhanced usability to ensure its effectiveness and adaptability in addressing adversarial AI threats.

Following are the technical (functional and non-functional) requirements identified for the AIAS xAI Mitigation Engine.

Table 16 Functional and non-functional requirements of the AIAS mitigation mechanism

		Description
S	ID	Req-Mitigation-FUNC-1
	Dependencies	USR-017
	Type	FUNC
	Short name	Real-Time Explainable Recommendations
	Description	The Mitigation Engine MUST provide real-time explainable recommendations for mitigating adversarial threats. These recommendations should be generated using XAI techniques like SHAP or LIME to ensure transparency and trust in the decision-making process. This feature is critical to enable human operators to quickly understand and act on the system’s suggestions, particularly in high-stakes environments where adversarial threats can escalate rapidly. The engine will also integrate with the AIDM for actionable insights and align with the Security Data Fusion component to leverage real-time data feeds.
	Additional Information	Recommendations must account for threat severity, contextual system data, and historical insights.
	Priority (MoSCoW)	M
M	Measures	Recommendation latency <2 seconds; operator trust >90% based on user feedback testing.

A	Achievable	Achievable through integration with XAI frameworks and real-time data pipelines.
R	Objectives	To provide actionable, understandable mitigation strategies that reduce the impact of adversarial attacks.
T	Timeline	Year 1: Develop basic recommendation engine and XAI integration. Year 2: Optimize latency and expand data sources. Year 3: Validate with pilot use cases and refine operator interfaces.
		Description
S	ID	Req-Mitigation-FUNC-2
	Dependencies	USR-018
	Type	FUNC
	Short name	Human-in-the-Loop Mitigation
	Description	The system SHOULD allow human operators to modify and approve mitigation actions suggested by the Mitigation Engine. This capability ensures alignment with organizational policies and enhances operator trust in automated systems. The feature will be implemented through an intuitive user interface integrated into the XAI-Based Mitigation Engine, enabling operators to review, validate, or override actions in real-time. This is particularly significant for complex scenarios where automated responses may not capture all nuances.
	Additional Information	The interface must integrate seamlessly with the AIAS platform and support decision logs for auditability.
	Priority (MoSCoW)	S
M	Measures	Operator approval/rejection accuracy >95%; average action review time <5 seconds.
A	Achievable	Achievable with UI/UX design optimized for security workflows and integration with the engine's decision logic.
R	Objectives	Enhance operational control while maintaining rapid response capabilities.
T	Timeline	Year 1: Prototype operator interface. Year 2: Full integration and operator feedback loops. Year 3: Advanced features like decision automation for low-risk scenarios.
		Description
S	ID	Req-Mitigation-FUNC-3
	Dependencies	USR-019
	Type	FUNC
	Short name	Historical Data Analysis

	Description	The Mitigation Engine MUST analyse historical data to ensure the effectiveness of proposed actions. This includes leveraging the Security Data Fusion component to access comprehensive historical datasets and employing advanced analytics to identify trends in adversarial tactics. Historical data analysis enables the engine to refine its mitigation strategies and avoid repeating ineffective measures, thereby improving its overall reliability and resilience.
	Additional Information	Requires integration with secure storage and processing pipelines for historical data.
	Priority (MoSCoW)	M
M	Measures	>85% accuracy in assessing the effectiveness of past mitigation strategies.
A	Achievable	Achievable through secure data integration and machine learning models trained on historical datasets.
R	Objectives	To enhance the engine’s decision-making capabilities through informed insights from past incidents.
T	Timeline	Year 1: Develop data access and initial analytics. Year 2: Expand historical datasets and refine analytics. Year 3: Validate results with pilot use cases and continuous improvement.
		Description
S	ID	Req-Mitigation-FUNC-4
	Dependencies	USR-020
	Type	FUNC
	Short name	Dynamic Learning from Adversarial AI Engine
	Description	The Mitigation Engine MUST dynamically learn from simulated attack scenarios generated by the Adversarial AI Engine. This integration allows the engine to continuously adapt to new adversarial tactics and refine its mitigation strategies. Such a capability is essential for maintaining the system’s relevance in the face of evolving threat landscapes. Real-time synchronization ensures the Mitigation Engine is equipped with the latest adversarial patterns.
	Additional Information	Requires API-based integration with the Adversarial AI Engine for seamless data exchange.
	Priority (MoSCoW)	M
M	Measures	Weekly updates to threat models; <5-second latency during simulation-based updates.
A	Achievable	Achievable through periodic synchronization protocols and machine learning model retraining.
R	Objectives	Enhance adaptability to emerging threats and ensure proactive mitigation.

T	Timeline	Year 1: Develop integration protocols. Year 2: Automate updates and retraining. Year 3: Optimize real-time synchronization.
		Description
S	ID	Req-Mitigation-FUNC-5
	Dependencies	USR-021
	Type	FUNC
	Short name	Probabilistic Scoring of Mitigation Strategies
	Description	The Mitigation Engine SHOULD provide probabilistic scoring for mitigation strategies, allowing operators to compare options based on their likelihood of success and potential risks. This feature supports transparency and informed decision-making, particularly in scenarios where multiple mitigation strategies are feasible. The scoring model will be informed by historical data, real-time threat intelligence, and contextual factors provided by the Security Data Fusion component.
	Additional Information	Must display scores in an operator-friendly format, such as graphical dashboards.
	Priority (MoSCoW)	S
M	Measures	Scoring accuracy >90%; operator satisfaction score >85% during testing.
A	Achievable	Achievable using Bayesian or reinforcement learning models for probabilistic analysis.
R	Objectives	Aid operators in selecting the most effective mitigation strategies for a given threat scenario.
T	Timeline	Year 1: Develop scoring algorithms. Year 2: Integrate with engine workflows. Year 3: Refine based on operator feedback.
		Description
S	ID	Req-Mitigation-FUNC-5
	Dependencies	USR-021
	Type	FUNC
	Short name	Probabilistic Scoring of Mitigation Strategies
	Description	The Mitigation Engine SHOULD provide probabilistic scoring for mitigation strategies, allowing operators to compare options based on their likelihood of success and potential risks. This feature supports transparency and informed decision-making, particularly in scenarios where multiple mitigation strategies are feasible. The scoring model will be informed by

		historical data, real-time threat intelligence, and contextual factors provided by the Security Data Fusion component.
	Additional Information	Must display scores in an operator-friendly format, such as graphical dashboards.
	Priority (MoSCoW)	S
M	Measures	Scoring accuracy >90%; operator satisfaction score >85% during testing.
A	Achievable	Achievable using Bayesian or reinforcement learning models for probabilistic analysis.
R	Objectives	Aid operators in selecting the most effective mitigation strategies for a given threat scenario.
T	Timeline	Year 1: Develop scoring algorithms. Year 2: Integrate with engine workflows. Year 3: Refine based on operator feedback.
		Description
S	ID	Req-Mitigation-NFUNC-1
	Dependencies	USR-017
	Type	PERFORMANCE
	Short name	Low Latency for Mitigation Recommendations
	Description	The Mitigation Engine MUST ensure that recommendations for mitigating adversarial threats are generated with minimal latency. This is critical for maintaining real-time responsiveness, especially in high-priority scenarios where delay could lead to significant damage. The system should achieve this by leveraging optimized computational resources, preloading frequently used threat models, and parallel processing techniques. Integration with the AI-Based Detection Module ensures real-time threat identification feeds directly into the recommendation pipeline.
	Additional Information	Latency targets must align with broader AIAS performance goals for real-time operations.
	Priority (MoSCoW)	M
M	Measures	95% of recommendations generated within <2 seconds.
A	Achievable	Achievable with high-performance computational infrastructure and efficient algorithm design.
R	Objectives	Ensure rapid mitigation response to minimize the impact of adversarial threats.
T	Timeline	Year 1: Develop latency benchmarks and prototype pipeline. Year 2: Optimize processing efficiency. Year 3: Validate low-latency performance in pilot use cases.

		Description
S	ID	Req-Mitigation-NFUNC-2
	Dependencies	USR-019
	Type	SECURITY
	Short name	Secure Data Integration
	Description	The Mitigation Engine MUST ensure the secure integration of data from historical datasets, real-time threat intelligence, and simulated scenarios. All data transfers between components (e.g., Security Data Fusion and Adversarial AI Engine) must use end-to-end encryption (e.g., TLS v.1.3) to prevent interception or unauthorized access. Additionally, data at rest must be encrypted using AES-256 standards to ensure compliance with the General Data Protection Regulation (GDPR) and other relevant regulations. This requirement ensures the confidentiality and integrity of sensitive threat intelligence.
	Additional Information	Periodic audits should verify adherence to data security standards.
	Priority (MoSCoW)	M
M	Measures	100% of data transfers encrypted; full compliance with GDPR and ISO 27001.
A	Achievable	Achievable using existing secure data transfer protocols and encrypted storage mechanisms.
R	Objectives	Safeguard sensitive data while enabling seamless integration across AIAS components.
T	Timeline	Year 1: Implement encryption protocols. Year 2: Test integration with components. Year 3: Validate against regulatory audits.
		Description
S	ID	Req-Mitigation-NFUNC-3
	Dependencies	USR-017, USR-020
	Type	RELIABILITY
	Short name	High Availability and Reliability
	Description	The Mitigation Engine MUST maintain high availability (99.9% uptime) and reliability to ensure continuous operation in mitigating adversarial threats. This requires redundancy mechanisms, such as failover systems and distributed deployments, to prevent service interruptions. Additionally, it should implement automated health monitoring to detect and address performance issues proactively. This ensures the system remains operational even under high loads or partial component failures, which is crucial for mission-critical environments.

	Additional Information	Includes automatic load balancing and resource scaling to handle peak demand.
	Priority (MoSCoW)	M
M	Measures	99.9% uptime; mean time to recovery (MTTR) <5 minutes during outages.
A	Achievable	Achievable using distributed architecture, cloud-based scaling, and automated monitoring tools.
R	Objectives	Ensure continuous availability to support real-time threat mitigation without interruptions.
T	Timeline	Year 1: Implement basic redundancy. Year 2: Introduce automated health monitoring. Year 3: Optimize failover and scaling mechanisms.

The five functional and three non-functional requirements for the Mitigation Engine collectively constitute the foundation of its capabilities within the AIAS architectural framework, addressing both core functionalities and critical operational parameters. The functional requirements define the engine's capacity to provide real-time recommendations that can be explained, enable human involvement in decision-making processes, analyse historical data, adapt dynamically to evolving threats, and offer probabilistic scoring for mitigation strategies. These requirements guarantee that the engine is not only responsive but also adaptive and transparent, thus making it integral to the AIAS architecture's overarching goal of defending against adversarial AI threats. To illustrate, the real-time recommendations and dynamic learning capabilities of the Adversarial AI Engine enable the system to maintain a proactive and relevant stance, while the incorporation of human-in-the-loop operations and probabilistic scoring enhances decision-making and operator trust. The incorporation of historical data analysis enables the engine to refine its mitigation strategies over time, thereby contributing to the development of a more robust defence framework.

The non-functional requirements serve to complement the aforementioned functionalities, thereby ensuring that the engine operates in an efficient, secure, and reliable manner. The generation of recommendations with minimal latency is of paramount importance for the real-time mitigation of threats, particularly in high-stakes scenarios where delays could result in substantial damage. The integration of sensitive threat intelligence across components, such as the Security Data Fusion and Adversarial AI Engine, is guaranteed through the implementation of secure data integration, which ensures the confidentiality and integrity of the data in accordance with regulatory standards, such as GDPR. Furthermore, high availability and reliability ensure that the Mitigation Engine operates continuously, even in the event of heavy load or partial system failure, thereby maintaining the resilience of the entire AIAS framework.

4.3.5 AIAS Security Data Fusion

The AIAS Security Data Fusion module will allow implementations of AIAS to follow a specific data sharing and exchanging mechanism, following a structured approach and providing a user interface for the user to inspect the integrated data. It will allow compliance with GDPR and Findable, Accessible, Interoperable, Reusable)(FAIR) principles.

User Requirements:

The following user-centred requirements ensure that users have a seamless experience when interacting with the system.

Table 17 User requirements of the AIAS Security Data Fusion component

Requirement ID	Requirement	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
USR-022	Various organisations COULD share knowledge (information of their data lakes)	C	Security Data Fusion
USR-023	Dashboard (custom or from a selected tool) to observe the data stored in the lake	S	Security Data Fusion
USR-024	Ability to explore the type of data that is being gathered by the Data Fusion Layer	M	Monitor and Security Analytics
USR-025	Ability to explore the data sources being used	M	Monitor and Security Analytics
USR-026	Ability to explore the attacks detected and how they are being used	S	Detection Mechanism
USR-027	All data assets must remain private when within the Data Fusion Layer	M	Security Data Fusion
USR-028	Compliance with ethics principles	M	Security Data Fusion
USR-029	Compliance with FAIR (Findable, Accessible, Interoperable, Reusable) principles	C	Security Data Fusion

Table 18 Functional and non-functional requirements of the AIAS Security Data Fusion component

S	ID	Req-Data_Fusion-FUNC-1
	Dependencies	Detection mechanism -- Data model of attacks
	Type	Functional
	Short name	Data Model Compliance
	Description	Compliance with the defined cyberattacks data model
	Additional Information	T3.4 output should be considered
	Priority (MoSCoW)	S: Should Have
M	Measures	Implementation and validation.
A	Achievable	100%
R	Objectives	The attacks that are detected are expressed in the defined format (T3.4) and stored likewise in the Data Fusion Module for exchange.
T	Timeline	M30/42/48
S	ID	Req-Data_Fusion-FUNC-2
	Dependencies	Detection mechanism + AI Adversarial Engine
	Type	Functional
	Short name	Data Prepared for Training
	Description	Data must be available for allowing training and inference of AI models
	Additional Information	Important to be validated by WP4.
	Priority (MoSCoW)	M: Must Have
M	Measures	Models in T4.2 and T4.3 can be fed by data exposed via Data Fusion module. XAI of T4.4 can be applied atop such models.
A	Achievable	100%
R	Objectives	The attacks that are detected are expressed in the defined format (T3.4) and stored likewise in the Data Fusion Module for exchange.
T	Timeline	M42
S	ID	Req-Data_Fusion-FUNC-3
	Dependencies	WP3 – Deception Layer
	Type	Functional
	Short name	Cybersecurity Provisions
	Description	It can include enough security provisions (Authorization, And Accounting (AAA)), either own or from other modules
	Additional Information	Deception Layer (functional) is considered a security provision for this requirement.

	Priority (MoSCoW)	C: Could Have
M	Measures	If implementations require additional cybersecurity elements, Data Fusion can incorporate authentication and authorization.
A	Achievable	100%
R	Objectives	The access to the Data handled by the Data Fusion Module has enough security protection.
T	Timeline	M39
S	ID	Req-Data_Fusion-FUNC-4
	Dependencies	None
	Type	Functional
	Short name	Batch or Stream Data Acceptance
	Description	It must accept both periodic and sporadic entries of data
	Additional Information	N/A
	Priority (MoSCoW)	M: Must Have
M	Measures	The technologies selected support both continuous data insertion (API o gRPC or Websocket or others) and periodic/sporadic batch loaders. The databases are as well prepared.
A	Achievable	100%
R	Objectives	Providing versatility to the way the cyberattacks detected are inserted into the Data Fusion Module.
T	Timeline	M36 (end of T4.1)
S	ID	Req-Data_Fusion-NFUNC-5
	Dependencies	None
	Type	Non-Functional
	Short name	Structured and non-structured data
	Description	Storage of structured and non-structured data about cybersecurity attacks.
	Additional Information	This requirement is complementary to Req-Data_Fusion-1 in the sense that the attacks can be expressed as defined in T3.4, but the Data Fusion module must accept any kind of input (informational input).
	Priority (MoSCoW)	M: Must Have
M	Measures	The technologies selected support both structured (e.g., in Structured Query Language (SQL)) and non-structured data.
A	Achievable	100%

R	Objectives	Providing versatility to the way the cyberattacks detected are inserted into the Data Fusion Module.
T	Timeline	M36 (end of T4.1)
S	ID	Req-Data Fusion- NFUNC-6
	Dependencies	None
	Type	Non-Functional
	Short name	API Exposure
	Description	Exposure of an API f to all modules of AIAS to retrieve information
	Additional Information	REST API is preferred.
	Priority (MoSCoW)	M: Must Have
M	Measures	The API of the Data Fusion Module is accessible and can be queried from the defined scope of the internal network of AIAS system.
A	Achievable	100%
R	Objectives	Working under standardised mechanisms to facilitate interoperability with the rest of AIAS architecture.
T	Timeline	M36 (end of T4.1)
S	ID	Req-Data Fusion- NFUNC-7
	Dependencies	None
	Type	Non-Functional
	Short name	API Exposure for data
	Description	Same as Req-Data_Fusion-6 but for inserting information (logs, statistics, attacks, etc.)
	Additional Information	REST API is preferred.
	Priority (MoSCoW)	M: Must Have
M	Measures	The API of the Data Fusion Module is accessible and can be queried from the defined scope of the internal network of AIAS system.
A	Achievable	100%
R	Objectives	Working under standardised mechanisms to facilitate interoperability with the rest of AIAS architecture.
T	Timeline	M36 (end of T4.1)
S	ID	Req-Data Fusion- NFUNC-8
	Dependencies	None

	Type	Non-Functional
	Short name	Open Source
	Description	Rooting on open source and available software
	Additional Information	REST API is preferred.
	Priority (MoSCoW)	M: Must Have
M	Measures	The selected technologies can be discovered in open-source Software repositories (at least 1).
A	Achievable	100%
R	Objectives	Compliance with best open research and science practices.
T	Timeline	M36 (end of T4.1)

4.3.6 Monitoring and analytics

The requirements of the AIAS monitoring and analytics tool are described below.

Table 19 Functional and non-functional requirements of the AIAS monitoring and analytics tool

S	ID	Req-Monitoring_and_analytics-FUNC-1
	Dependencies	N/A
	Type	Functional
	Short name	Data collection
	Description	The tool must collect data from diverse sources (network logs, endpoint data, application logs, etc.) in real time.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Retrieve data in real time mode <100 seconds
A	Achievable	100%
R	Objectives	Collect data from various sources.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-2
	Dependencies	N/A
	Type	Functional
	Short name	Data format
	Description	The tool must receive multiple data formats for logs and events.
	Additional Information	None

	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Implementation and validation.
A	Achievable	Support at least two different data structure formats.
R	Objectives	Receive multiple data formats for logs and events.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-3
	Dependencies	N/A
	Type	Functional
	Short name	Data volume
	Description	The tool must handle large volumes of data.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Response in less than 50 seconds
A	Achievable	100%
R	Objectives	Handle large volumes of data.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-4
	Dependencies	N/A
	Type	Functional
	Short name	Harmonize
	Personas	N/A
	Description	The tool must harmonize logs from different sources to a common data scheme in case of a need.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Harmonize in case of a need in less than 60 seconds
A	Achievable	100%
R	Objectives	Harmonize logs from different sources.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-5
	Dependencies	N/A

	Type	Functional
	Short name	Data correlation
	Description	The tool must be able to correlate security events across multiple data sources for holistic threat analysis.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Integrate at least 2 rules.
A	Achievable	100%
R	Objectives	Correlate data with predefined rules/thresholds.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-6
	Dependencies	N/A
	Type	Functional
	Short name	Patterns analysis
	Description	The tool must detect anomalies, suspicious patterns, and violations of predefined rules.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Anomaly identification in less than 100 seconds
A	Achievable	100%
R	Objectives	Detect anomalies, suspicious patterns, and violations of predefined rules.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-7
	Dependencies	N/A
	Type	Functional
	Short name	Alerts
	Description	The tool must generate real-time alerts based on defined thresholds, patterns, and anomalies.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Deliver aggregative report in less than 5 minutes.
A	Achievable	100%

R	Objectives	Generate real-time alerts.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-8
	Dependencies	N/A
	Type	Functional
	Short name	Visualization
	Description	The tool must provide intuitive, customizable dashboards for visualizing key metrics
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	UI response in less than 10 seconds.
A	Achievable	100%
R	Objectives	Support a customizable dashboard.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-9
	Dependencies	N/A
	Type	Security
	Short name	Anonymization
	Description	The tool must support data anonymization techniques to protect sensitive data.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Implementation and validation.
A	Achievable	100%
R	Objectives	The processed data must be anonymized.
T	Timeline	M20/42/48
S	ID	Req-Monitoring_and_analytics-FUNC-10
	Dependencies	N/A
	Type	Security
	Short name	API

	Description	The tool must support secure APIs for seamless integration with other AIAS-tools.
	Additional Information	None
	Priority (MoSCoW)	M: Must-have. Mandatory requirement.
M	Measures	Use 1 secure API to communicate with all other components.
A	Achievable	100%
R	Objectives	Integrate an API to communicate with other AIAS-tools.
T	Timeline	M20/42/48

Table 20 User requirements of the AIAS monitoring and analytics tool

Requirement ID	**Requirement** (one line definition)	**Priority** (**M**: Must-have. Mandatory Requirement, **S**: Should-have. Desirable Requirement, **C**: Could-have. Optional Requirement, **W**: Will-not-have. Possible Future Enhancement)	**Architecture Component** (that could address the requirement)
USR-030	Users must be able to customize dashboards to display relevant metrics, logs, and security events.	M	Monitoring and analytics tool
USR-031	Users must be able to search and analyse historical security logs and events to identify trends, investigate past incidents, and ensure compliance.	M	Monitoring and analytics tool
USR-032	The interface must be intuitive and user-friendly, ensuring that even non-technical users can navigate dashboards, configure alerts, and generate reports easily.	M	Monitoring and analytics tool

5 Reference Architecture

The AIAS architectural design represents a sophisticated cybersecurity solution tailored for SMEs, with a particular focus on countering adversarial AI threats and ensuring resilient AI operations. The AIAS platform's fundamental components include an adversarial AI engine, deception mechanisms, detection and mitigation modules, and a robust data fusion framework.

The adversarial AI engine has been designed to model attack scenarios that are specifically aligned with the unique hardware and software configurations that are typically found within SMEs. By generating and executing simulated attacks, the engine is able to identify and expose system vulnerabilities, thus facilitating the development of tailored defences. Furthermore, the deception layer utilises sophisticated techniques, including high-interaction honeypots, digital twins and virtual personas, which imitate the operational environment of the SME in question. This virtual layer is designed to effectively trap and analyse malicious activities, thereby diverting attackers away from critical systems while gathering intelligence to refine defence strategies. The AIAS detection module employs lifelong reinforcement learning, enabling it to continuously adapt to emerging threats and improve detection accuracy without requiring constant retraining from scratch. With regard to defence, the mitigation module is particularly innovative; it employs XAI to provide SMEs with clear, actionable insights on responding to cyber threats. These XAI-driven recommendations rely on a human-in-the-loop approach, combining machine and human decision-making in a transparent and accessible manner.

In addressing the significant challenges faced by small and medium-sized enterprises in the context of limited resources, a lack of technical cybersecurity expertise, and the rapid evolution of adversarial AI, the AIAS platform's layered approach provides a defence that is both cost-effective and technically sophisticated. The system proactively safeguards AI systems by addressing vulnerabilities such as data poisoning, evasion attacks and model theft. The platform's data fusion layer, which aggregates security data from a variety of SME installations, provides an additional layer of robustness through continuous learning and adaptation to the broader cybersecurity ecosystem. Moreover, the interdisciplinary nature of AIAS, which draws upon insights from cybersecurity, AI, and digital forensics, serves to reinforce its relevance as a comprehensive and adaptive cybersecurity solution. In this way, AIAS guarantees that SMEs are able to safeguard their AI-based systems in accordance with their operational limitations, thereby maintaining system confidentiality, integrity, and availability in the context of adversarial threats. By uniting the knowledge of industry experts with that of academics, AIAS establishes a new benchmark in cybersecurity resilience, particularly for SMEs with limited resources that must navigate the increasingly sophisticated digital threats they face.

5.1. AIAS Architecture principles

The AIAS architecture represents a comprehensive and modular cybersecurity framework, designed to provide proactive protection, detection, and mitigation for AI-based systems within SME environments. This architectural framework is founded upon principles derived from the resilience of adversarial artificial intelligence, the utilisation of deception technologies, the concept of continuous learning, and the provision of explainable decision support. The various components of the AIAS architecture collectively constitute a multilayered defence system that enhances the ability of SMEs to withstand and counter sophisticated cyber

threats while imposing minimal technical overhead. The modular nature of the architecture is a crucial factor in ensuring scalability and flexibility. It enables organisations to integrate AIAS seamlessly into their existing infrastructure while addressing specific cybersecurity challenges.

5.2. Proactive Defence through Adversarial Simulation

The fundamental tenet of AIAS's architectural framework is its proactive strategy for safeguarding against adversarial AI and cybersecurity threats through the simulation of potential attacks. The Adversarial AI Engine exemplifies this principle by generating sophisticated attack scenarios tailored to the specific configurations of SME systems. By establishing a systematic classification of adversarial attack vectors, encompassing elements such as algorithmic approaches, hyperparameters, and training data, the engine is able to identify vulnerabilities in a methodical manner. Deep neural networks, in particular generative adversarial networks (GANs), are employed in conjunction with attack graph methods to construct these scenarios, thereby enabling comprehensive testing of the system's resilience. This pre-emptive simulation enables AIAS to anticipate and prepare for emerging adversarial tactics, thereby ensuring its readiness to defend against both traditional cyber threats and adversarial AI attacks, such as evasion and poisoning attacks.

5.3. Layered Defence with Deception Technologies

A second fundamental principle of the AIAS architectural design is the implementation of a layered defence strategy, which aims to contain potential threats at multiple points of potential compromise. The Deception Layer represents the initial point of contact for potential adversaries, offering a range of sophisticated tools, including high-interaction honeypots, digital twins, and virtual personas that closely resemble the operational environment of the SME. The objective of this deception layer is to absorb and analyse malicious behaviour without affecting the operational infrastructure. Digital twins mirror the organisation's assets and workflows, while virtual personas replicate user behaviour, effectively deceiving attackers into believing they are interacting with the genuine system. This interaction serves two purposes: Firstly, it diverts potential threats, and secondly, it gathers valuable data on attacker Tactics, Techniques, and Procedures (TTPs). This data is then analysed by the platform's security analytics. The insights gathered are fed into other AIAS modules, strengthening their threat prediction and mitigation capabilities. This approach can be described as a "security by isolation" approach, whereby malicious activities are contained in an isolated layer.

5.4. Continuous Learning and Adaptation

The AIAS architectural design incorporates LLRL within its detection module, thereby enabling continuous and dynamic adaptation to emerging threats. The AI-based Detection Module (AIDM) has been designed to evolve in response to both real-time attack data and simulated adversarial scenarios. The LLRL approach employs continuous feedback loops, drawing data from the adversarial AI engine and deception layer to refine its anomaly detection models.

In order to optimise the learning process, the AIDM employs a range of sophisticated data processing techniques, including feature selection, dimensionality reduction and unsupervised clustering. These techniques are designed to extract high-value patterns from complex data sets. This continuous learning principle enhances the ability of AIAS to detect novel and zero-day attacks with high accuracy and low

latency, enabling a rapid response to evolving threats. This principle is of particular importance for SMEs, where rapid adaptation can minimise the impact of attacks on limited resources and infrastructure.

5.5. Explainability and Human-Centric Decision Support

A fundamental tenet of the AIAS architectural framework is the provision of transparency and explainability in its recommendations. This is achieved through the deployment of an XAI-based Mitigation Engine. In light of the necessity for human involvement in cybersecurity, particularly in SMEs with constrained cybersecurity resources, the XAI component furnishes security operators with transparent and comprehensible justifications for recommended actions. The mitigation engine employs the use of SHAP and LIME techniques so as to elucidate the logic behind the mitigation suggestions that it makes. By employing an "*if-this-then-that*" methodology, the XAI-driven module assists decision-makers in selecting optimal responses, thereby enhancing trust in the system and facilitating effective human-in-the-loop operations. This principle guarantees that SMEs, even with restricted in-house cybersecurity expertise, can utilise the AIAS platform to make well-informed and efficient decisions, thereby enhancing their cybersecurity preparedness.

5.6. Secure, Decentralized, and Collaborative Data Management

The AIAS architectural framework is founded upon the principles of decentralised data management and collaborative knowledge sharing, with the objective of enhancing the resilience of its defensive capabilities across a range of SME implementations. The “Security Data Fusion” component collates security-related data from log files, network traffic and the results of both simulated and real attack events. The data is stored in a federated data structure using IPFS and Hyperledger Fabric, thereby providing a decentralised and tamper-resistant environment. By adhering to the principles of GDPR and the FAIR data principles, AIAS ensures the secure and privacy-preserving management of data, while enabling interoperability across organisational boundaries. Furthermore, the Decentralized Knowledge Base gathers anonymized threat intelligence from multiple instances of AIAS, thereby establishing a shared repository of attack signatures and defence strategies. This principle facilitates mutual defence, allowing SMEs to benefit from each other's experiences with adversarial attacks and thereby foster a collaborative cybersecurity ecosystem.
As an initial attempt to define the collaborative data management, AIAS’s AIDM will utilise the knowledge extracted from the deception and monitoring tools and the generated adversarial AI attack scenarios to develop novel powerful detection and mitigation techniques. The AIDM will utilise a life-long reinforcement learning approach to detect anomalies on the system continuously and dynamically. Moreover, AIAS will research and implement a data lake that will collect, process, and fuse security data at an organization level (Security Data Fusion) at an organization level, which will be used to train the AIAS AI models (e.g., adversarial AI engine, AIDM). A decentralized knowledge base (AIAS Decentralized Knowledge Base) will be developed to gather data from various organizations complying with GDPR based on the AIAS’s privacy impact assessment.

For achieving so, a first design has been produced:

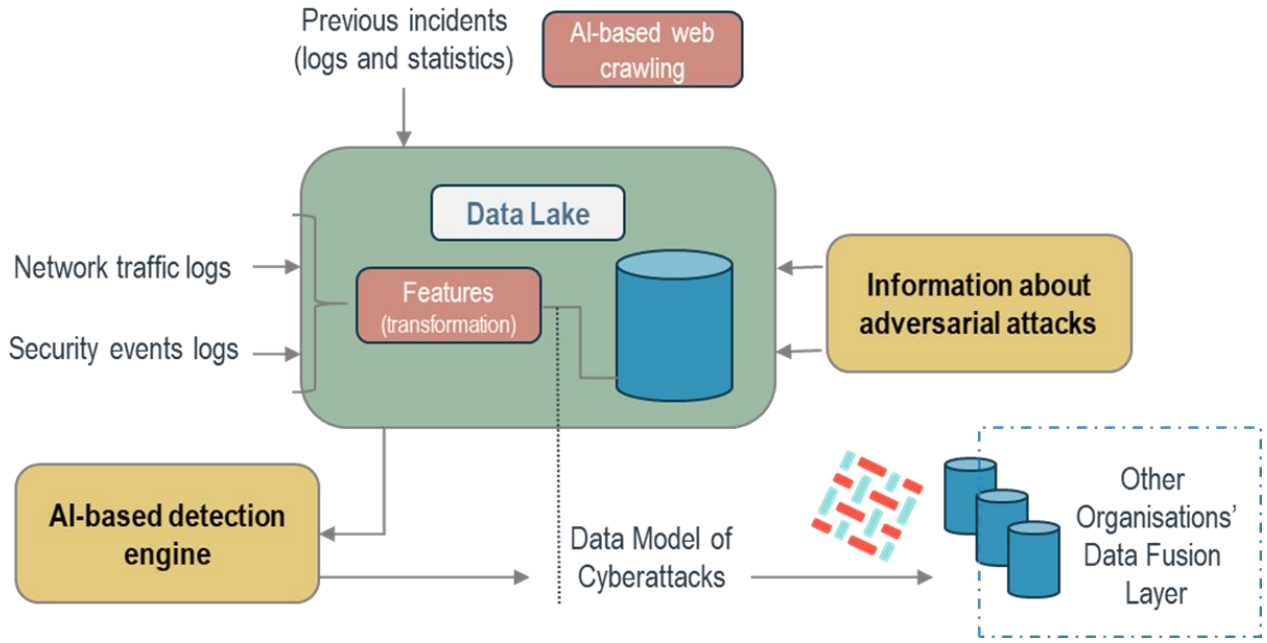

Fig. 3 Initial design of AIAS AI-based Data Fusion Module for detection

5.7. Scalability and Modularity for Adaptability

The AIAS architectural design is inherently scalable and modular, thereby enabling SMEs to integrate the platform into their existing systems with minimal disruption. The modular design of the AIAS architecture allows for the selective implementation of specific components, such as the detection module or the deception layer, in accordance with the particular needs and resources of the organisation in question. This adaptability is of particular importance for SMEs, which frequently operate under budgetary and infrastructure constraints. The modular structure also facilitates the integration of future updates or additional AIAS features, thereby enabling the system to evolve in accordance with advancements in adversarial AI tactics and defences. Furthermore, the scalability of AIAS is enhanced by its data fusion and decentralised knowledge base, which ensure that threat intelligence and detection capabilities can be expanded to accommodate additional sources and new types of data without requiring significant architectural modifications.

5.8. AIAS Architecture description

The AIAS architectural framework is constituted by an integrated set of components, each of which is designed to contribute to the formation of a unified cybersecurity defence system capable of safeguarding SMEs from sophisticated adversarial AI and cyber threats. The architecture is modular, thereby enabling the deployment of individual components while maintaining interoperability across the system. The system comprises several key components, including the Adversarial AI Engine, the Deception Layer, the AI-based Detection Module, the XAI-based Mitigation Engine, and the Security Data Fusion and Decentralised Knowledge Base. Each of these components fulfils a distinct role within the AIAS platform. These include the simulation of adversarial attacks and the gathering of intelligence through the utilisation of sophisticated

deception technologies, the continuous adaptation of detection models using reinforcement learning, and the facilitation of human-in-the-loop decisions with the incorporation of XAI insights.

From a technical standpoint, AIAS utilises sophisticated methodologies, including GANs and attack graphs, within its adversarial engine to simulate and stress-test AI systems. Additionally, it employs high-interaction honeypots, digital twins, and virtual personas in the deception layer to capture and analyse malicious interactions. Communication and data flow between components are managed through the utilisation of standardised protocols and interfaces, thereby ensuring seamless data integration and real-time responsiveness across the platform. The Security Data Fusion component employs a decentralised and federated storage system, implemented via IPFS and Hyperledger Fabric, to ensure the integrity and privacy of data while facilitating the sharing of threat intelligence across multiple instances of AIAS. This chapter provides a comprehensive account of the technical design of each component, delineating its distinctive functions, mechanisms, protocols, and interfaces. This is done to demonstrate how AIAS, in its entirety, constitutes a robust, flexible, and SME-oriented cybersecurity solution.

5.8.1 AIAS Adversarial AI Engine and Deception

Within the AIAS architecture, the Adversarial AI Engine Module (AI2EM) serves a dual purpose. Firstly, it is responsible for generating adversarial AI attacks, including Poisoning [RKH], Evasion [EAV], and Transfer attacks [MFW], which are strategically directed towards the AIAS deception layer. Secondly, AI2EM constructs Attack Graphs—structured representations that model potential security threats and vulnerabilities—highlighting pathways through which various vulnerabilities might be exploited. To this end, AI2EM is composed of three core sub-modules: 1) The Weaponizer, 2) Deep Neural Networks (DNNs), and 3) A Taxonomy of Adversarial AI Attacks. In Fig. 4, these three sub-modules are presented as the fundamental components of the AI2EM module.

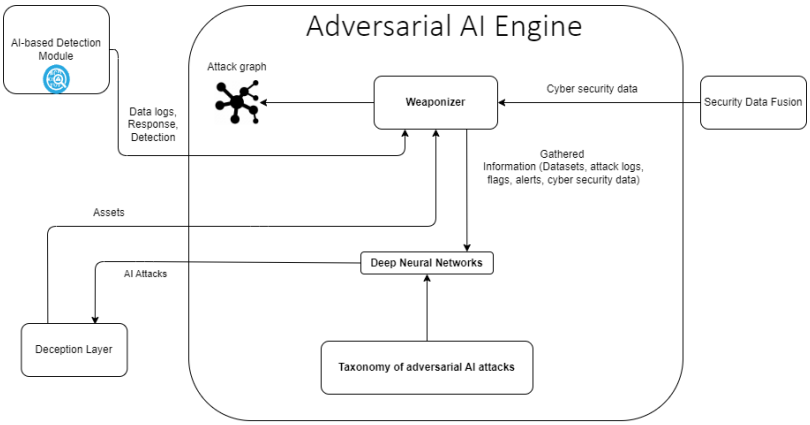

Fig. 4 Architecture of the Adversarial AI Engine Module

To synthesize and implement adversarial attacks and attack graphs, the AI2EM module and its sub-modules require data from several key components within the AIAS architecture. These components include:

- ***The Deception Layer***: A module that generates a virtual imitation of an organization, designed to deceive adversaries by emulating the organization’s environment and behaviour.

- ***The AI-based Detection Module***: This module leverages reinforcement learning techniques to detect both cyberattacks and AI-specific adversarial attacks within the system.
- ***The Security Data Fusion***: A centralized data repository (or data pool) that aggregates security-related data from multiple sources. This includes records of attacks detected by the Deception Layer, events captured by the AI-based Detection Module, and adversarial AI attacks generated by AI2EM. For more details, refer to Subsection 2.5.

Before delving into the primary components of AI2EM, it is beneficial to examine in greater detail the submodules that provide critical data inputs to AI2EM. Gaining an understanding of these supporting modules will offer a clearer picture of how they contribute to AI2EM’s functionality. To begin, we will analyse the architecture of the Deception Layer.

The Deception Layer functions by constructing a virtual replica of the organization’s Information Communication Technology (ICT) infrastructure, designed to mislead adversaries and attract potential attacks on the organization’s AI models and AI-driven systems. To fulfil this purpose, the Deception Layer integrates a deception mechanism, utilizes security analytics to process data collected through these mechanisms, and deploys network monitoring tools.

The deception mechanism itself consists of a High Interaction Honeypot and an Advanced High Interaction Honeypot. The latter represents a novel development within the AIAS project, combining Digital Twins, Virtual Personas, and High Interaction Honeypots. This integrated approach offers attackers a highly realistic simulation of the organization, creating a credible target to enhance the effectiveness of the deception strategy. In Fig. 5, we observe the submodules that comprise the Deception Layer.

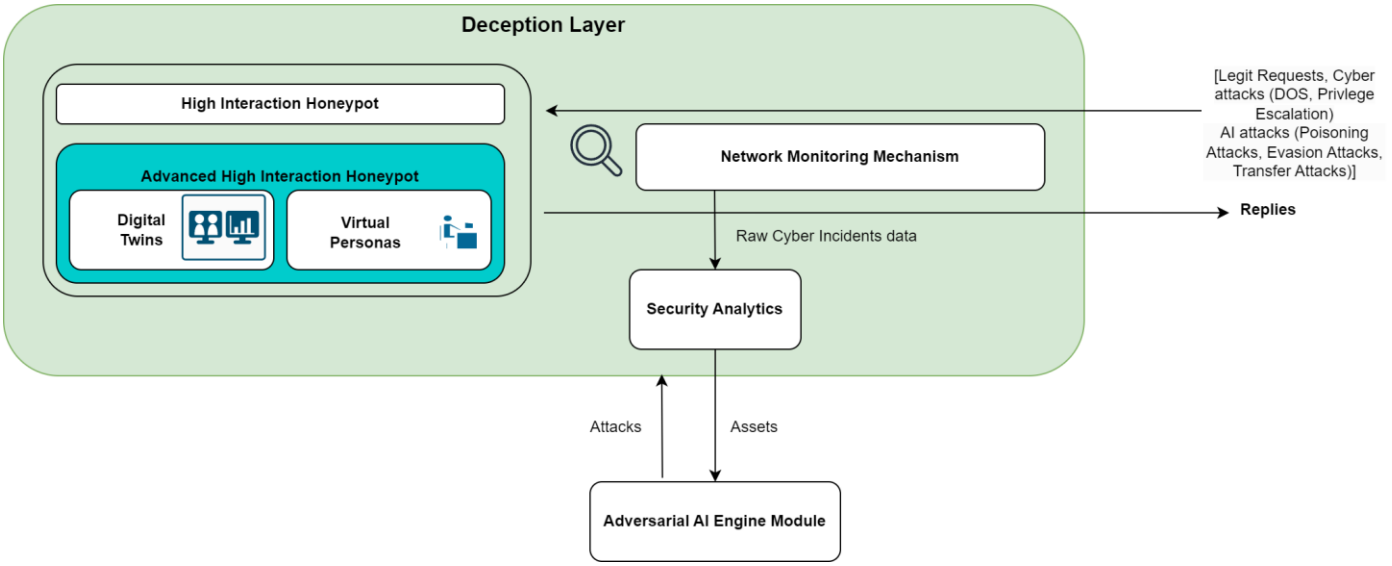

Fig. 5 Architecture of the Deception Layer

Following, we will analyse the submodules of the Deception Layer in more detail, highlighting the role each component plays in constructing a realistic and compelling deception environment for adversaries.

- ***Digital Twins***: Digital Twins are real-time, digital replicas of physical objects, systems, and processes. They are designed to mirror the functions and operations of their real-world counterparts, with continuous updates from sensors or other data sources (e.g., network traffic) to accurately reflect the evolving state and behaviour of the physical entity.
- ***Virtual Personas***: Virtual Personas are digital representations of individual identities, primarily used in online platforms or virtual environments. They mimic the actions and behaviours of real individuals, though not necessarily in real time, providing an additional layer of realism within the virtual deception environment.
- ***High Interaction Honeypots***: High Interaction Honeypots offer a genuine operating system environment that appears vulnerable to external threats, making it an attractive target with a high probability of engaging a human adversary. These honeypots yield extensive information on an attacker’s activities, as attackers often invest significant time and effort attempting to exploit perceived vulnerabilities within these systems.

Now that we have analysed the submodules, we proceed to examine how the deception mechanism operates as a cohesive system to lure and engage adversaries effectively.

The goal of the deception mechanism is to create a holistic imitation, not only of the organization’s ICT infrastructure but also of typical human behaviours. This realistic setup attracts attackers by presenting what appears to be a genuine environment, aimed at deceiving both human and automated adversaries.

In this setup, the two honeypots play a central role. These honeypots actively receive requests from both external internet sources and the organization’s internal network. Incoming requests may include legitimate traffic as well as malicious attacks, which could take forms such as Denial of Service (DoS), Distributed DoS, Privilege Escalation, and AI-specific attacks like Poisoning, Evasion, and Transfer attacks. In response, the honeypots generate convincing replies to these requests, effectively tricking adversaries into believing they are interacting with a legitimate system.

To support this interaction, the Network Monitoring Mechanism continuously tracks all incoming and outgoing traffic. This mechanism includes tools such as an IDS, packet sniffing tools, agent-based monitoring, or a combination of these, providing a comprehensive view of network activity. Data collected through these tools—alerts, flags, and logs—is forwarded to the Security Analytics module, where it is processed and analysed. The analytics module then visualizes the data, presenting it in a way that is accessible and interpretable by security analysts.

Once the data has been processed, it is fed into the AI2EM module, which utilizes this information alongside inputs from other modules to synthesize adversarial AI attacks. These AI-driven attacks are subsequently launched back against the Deception Layer, mirroring real-world attack patterns and contributing to a continuous, dynamic deception strategy. In this way, AI2EM-generated attacks become part of the requests that engage with the Deception Layer, enhancing the realism and effectiveness of the entire system.

By following this coordinated flow, the AIAS architecture creates an engaging deception environment that is continually adaptive, capable of deceiving a wide range of adversaries, and equipped to capture invaluable data on attacker behaviours.

With the analysis of the Deception Layer complete, we now return to the AI2EM module to explore its primary components in greater depth. This includes an examination of the Weaponizer, the DNNs, and the Taxonomy of Adversarial AI Attacks, each of which plays a crucial role in synthesizing and implementing adversarial AI strategies within the AIAS architecture.

The ***Weaponizer*** serves as the initial processing unit within the AI2EM module, continuously receiving cybersecurity-related data from the Security Data Fusion module, data logs from the AI-based Detection Module, and information on organizational assets from the Deception Layer. It preprocesses this information to prepare it for subsequent stages, transforming the raw data into a suitable format before passing it to the DNNs.

The ***DNNs*** operate as the core analytical engine within AI2EM. In addition to receiving pre-processed data from the Weaponizer, the DNNs are also fed information from the ***Taxonomy of Adversarial AI Attacks***, which provides a structured classification of various attack types. With both data inputs, the DNNs generate and refine AI-based attacks, which are then directed toward the Deception Layer. The DNNs fulfil several key functions:

- **Generation of Adversarial Examples**: Crafting adversarial samples to test the robustness of AI models within the Deception Layer.
- **Optimization of Attack Strategies**: Learning and adapting attack methods by identifying patterns in the target model’s behaviour and operation.
- **Automation and Transferability of Attacks**: Streamlining the attack process to adapt dynamically, enhancing the transferability of attack strategies.
- **Defence Bypass**: Identifying and circumventing specific defensive mechanisms, allowing the attacks to penetrate defences effectively.
- **Creation of Sophisticated Attacks**: Producing complex and multi-layered attacks that closely mimic real-world adversarial threats.
- **Adversarial attack Identification:** It differentiates between normal and drifted data by analyzing incoming inputs to determine if they result from adversarial actions or align with the model’s training. This functionality relies on sophisticated detection algorithms to identify characteristics of adversarial data.
- **Model Resilience Testing:** By exposing the model to adversarially generated data, this function evaluates the model’s ability to maintain performance under manipulated conditions. This process helps identify vulnerabilities and strengthen the model’s defenses against real-world threats.

Together, the Weaponizer and DNNs, supported by the Taxonomy of Adversarial AI Attacks, drive the AI2EM module’s capacity to produce targeted, adaptive, and effective adversarial AI attacks against the organization’s defences.

5.8.2 AIAS AI-driven Detection and Mitigation

The AI-driven detection and mitigation technologies include the User and Entity Behaviour Analytics (UEBA) and Intrusion Detection and Prevention Systems (IDPS). These technologies leverage artificial intelligence to detect anomalies and mitigate potential threats effectively. The architecture consists of the Data Sources,

which include logs from applications, servers, network devices, and real-time network traffic.

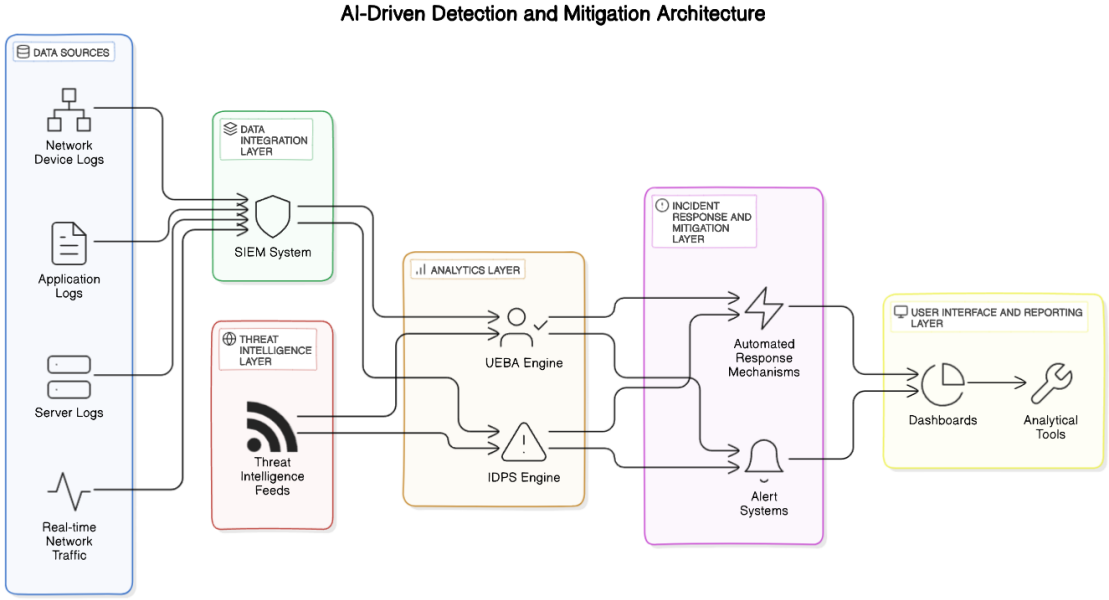

Fig. 6 AI-Driven Detection and Mitigation Architecture

The Data Integration Layer, featuring the Security Information and Event Management (SIEM) system that aggregates and normalizes data for analysis. The Analytics Layer, where the UEBA engine analyses user behaviour to establish baselines and detect anomalies, while the IDPS engine monitors network traffic for suspicious patterns. The Threat Intelligence Layer provides contextual information about known threats to improve detection capabilities. The Incident Response and Mitigation Layer, coordinating responses to detected threats through automated mechanisms and alert systems. The User Interface and Reporting Layer, facilitates monitoring and reporting through dashboards and analytical tools. This architecture collectively strengthens an organization’s ability to identify, respond to, and mitigate cybersecurity risks effectively.

UEBA integrates with the SIEM system to enhance threat detection and response capabilities within an organization. SIEM systems collect and aggregate data from various sources, including logs from servers, applications, and network devices. This raw data serves as the foundation for security monitoring. UEBA enhances this by analysing the collected data to identify behavioural patterns and anomalies that may indicate security risks. This integration allows security teams to leverage the strengths of both technologies, providing a more comprehensive view of user activities and potential security threats. UEBA focuses on analysing the behaviour of users and entities within a network. By establishing a baseline of normal behaviour, UEBA systems can identify deviations that may indicate malicious activities or security breaches. Logs are crucial in UEBA as they provide the data needed for behavioural analysis.

Logs from various sources, such as servers, applications, and network devices, are collected to create a comprehensive view of user activities. UEBA systems collect logs from a variety of sources, including servers, applications, network devices, and user activity logs. This comprehensive data gathering is essential for understanding the typical behaviour of users and entities in the network Historical log data is analysed to establish what constitutes 'normal' behaviour for users and entities. The collected log data is analysed to create

a baseline profile of normal behaviour for each user and entity. This involves tracking various activities such as login times, file access patterns, and system interactions over time. For example, if a user typically logs in at 9 AM and accesses specific files, these patterns become part of their behavioural profile. Once the baseline is established, UEBA continuously monitors real-time log data for deviations from this baseline. Any significant deviation—such as a user accessing the system at an unusual hour or downloading an atypical volume of data—triggers an alert. This is achieved through machine learning algorithms that refine the baseline over time, improving the accuracy of anomaly detection.

UEBA assigns risk scores to detected anomalies based on their severity and potential threat level. For instance, multiple failed login attempts or unusual access to sensitive files may result in a higher risk score, prompting immediate investigation by security analysts. This scoring system helps prioritize alerts, allowing teams to focus on the most critical threats first. To enhance the investigation process, UEBA systems supplement alerts with contextual information derived from logs. This includes details about user privileges, historical behaviour patterns, and the nature of the anomaly itself. Such context aids analysts in understanding the potential implications of the detected abnormal behaviour.

Integrating threat intelligence feeds improve the identification and assessment of emerging threats based on real-time data and historical patterns. This integration allows for the correlation of incoming security events with known indicators of compromise, enabling rapid detection and informed responses to potential incidents. Additionally, threat intelligence enriches the data processed by the SIEM and UEBA, facilitating more accurate anomaly detection and reducing false positives. It also supports incident response efforts by delivering actionable insights during security incidents, helping teams quickly identify the nature and scope of threats.

Continuous monitoring of logs allows the system to detect anomalies by comparing real-time activities against the established baseline. For example, if a user typically logs in during business hours but suddenly accesses the system at midnight from a different location, this could trigger an alert. Incident Response: When anomalies are detected, UEBA systems can initiate automated responses or alert security personnel for further investigation.

IDPS are critical components of network security that monitor traffic for suspicious activities. The integration of AI enhances their capabilities significantly. AI-powered IDPS can analyse network traffic patterns to identify abnormal flows that may signify an intrusion or attack. AI algorithms analyse incoming and outgoing traffic in real-time, looking for patterns that deviate from normal behaviour. These systems use ML models trained on historical data to recognize legitimate versus malicious traffic. For instance, if a sudden spike in data transfer occurs from a device that usually has low activity, it raises flags. Upon detecting abnormal flows, AI-driven IDPS may automatically take actions such as blocking suspicious IP addresses or isolating affected systems to prevent further damage.

6 Conclusions

Deliverable 2.1 is essential for the subsequent phases of AIAS project for multiple reasons: (i) It defines the user, functional and non-functional requirements of each AIAS component; and (ii) specifies a technical description of each AIAS functional component. The AIAS architecture delineates the system modules, and the technologies employed for intercommunication, while also considering the system needs. AIAS architecture considers the aforementioned technologies to accomplish its objectives by delivering an integrated security platform. The reference architecture serves as the foundation for the design and execution of technological solutions for AIAS.

References

[VLR] Volere Requirements: How to Get Started, http://www.volere.co.uk/pdf%20files/VolereGettingStarted.pdf

[SSE] ISO/IEC. 2007. Systems and Software Engineering -- Recommended Practice for Architectural Description of Software-Intensive Systems. Geneva, Switzerland: International Organization for Standards (ISO)/International Electrotechnical Commission (IEC), ISO/IEC 42010:2007

[INC] INCOSE. 2010. Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. Version 3.2.1. San Diego, CA, USA: International Council on Systems Engineering (INCOSE), INCOSE-TP- 2003-002-03.2.1: 362

[SMR] Doran, G. T. (1981). There's a SMART way to write management's goals and objectives. *Management review*, *70*(11).

[MSC] MoSCoW Prioritisation, Agile business Consortium, Online: https://www.agilebusiness.org/dsdm-project-framework/moscow-prioririsation.html [Last access: 25/9/2024]

[QLZ] Qiu, S., Liu, Q., Zhou, S., & Wu, C. (2019). Review of artificial intelligence adversarial attack and defense technologies. Applied Sciences, 9(5), 909.

[CLD] CALDERA™ is a cyber security platform Online: https://github.com/mitre/caldera [Last access: 25/9/2024]

[ATR] Atomic Red Team™ Online: https://github.com/redcanaryco/atomic-red-team [Last access: 25/9/2024]

[MAM] MITRE ATT&CK Mitigation Techniques Online: https://attack.mitre.org/mitigations/enterprise/ [Last access: 25/9/2024]

[CIS] Center of Internet Security Online: https://www.cisecurity.org/ [Last access: 25/9/2024]

[SCF] Secure Controls Framework, Security & Privacy Metaframework, Online: https://www.securecontrolsframework.com [Last access: 25/9/2024]

[AJF] Javadpour, Amir, et al. "A comprehensive survey on cyber deception techniques to improve honeypot performance." *Computers & Security* (2024): 103792.

[HPY] https://github.com/foospidy/HoneyPy

[IFL] C. Irvene, D. Formby, S. Litchfield and R. Beyah, "HoneyBot: A Honeypot for Robotic Systems," in Proceedings of the IEEE, vol. 106, no. 1, pp. 61-70, Jan. 2018, doi: 10.1109/JPROC.2017.2748421.

[SWK] C. Seifert, I. Welch, and P. Komisarczuk, "HoneyC: The Low-Interaction Client Honeypot,", 2006. http://www.mcs.vuw.ac.nz/~cseifert/blog/images/seifert-honeyc.pdf

[RRM] M. F. Razali, M. N. Razali, F. Z. Mansor, G. Muruti and N. Jamil, "IoT Honeypot: A Review from Researcher's Perspective," 2018 IEEE Conference on Application, Information and Network Security (AINS), Langkawi, Malaysia, 2018, pp. 93-98, doi: 10.1109/AINS.2018.8631494.

[HAU] M. A. Hakim, H. Aksu, A. S. Uluagac and K. Akkaya, "U-PoT: A Honeypot Framework for UPnP-Based IoT Devices," 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, FL, USA, 2018, pp. 1-8, doi: 10.1109/PCCC.2018.8711321.

[COW] Cowrie, Online, https://www.cowrie.org/ , Last accessed on 10-12-2024

[DIO1] Seguridara, PoC: Captura de malware con el honeypot Dionaea - Parte I https://revista.seguridad.unam.mx/numero23/poc-captura-de-malware-con-el-honeypot-dionaea-parte-i [Online] [Last access 11/12/2024]

[DIO2] Dionaea, https://dionaea.readthedocs.io/en/latest/introduction.html, [Online] [Last access 11/12/2024]

[GLA] Glastopf https://github.com/mushorg/glastopf [Online] [Last access 11/12/2024]

[GLS] Glastopf: Honeypot de aplicaciones web – I https://revista.seguridad.unam.mx/numero25/glastopf-honeypot-de-aplicaciones-web-i [Online] [Last access 11/12/2024]

[MHN] HoneyDrive https://github.com/pwnlandia/mhn [Online] [Last access 11/12/2024]

[HDR] https://sourceforge.net/projects/honeydrive/ [Online] [Last access 11/12/2024]

[HTH] HoneyThing https://github.com/omererdem/honeything [Online] [Last access 11/12/2024]

[CON] CONPOT ICS/SCADA Honeypot http://conpot.org/ [Online] [Last access 11/12/2024]

[KIP] Kippo honeypot, https://github.com/desaster/kippo [Online] [Last access 11/12/2024]

[HSS] Honssh, https://github.com/tnich/honssh [Online] [Last access 11/12/2024]

[MNT] Kaggle, MNIST Dataset, https://www.kaggle.com/datasets/hojjatk/mnist-dataset [Online] [Last access 11/12/2024]

[INET] ImageNet https://www.image-net.org/ [Online] [Last access 11/12/2024]

[MFW] Mao, Y., Fu, C., Wang, S., Ji, S., Zhang, X., Liu, Z., ... & Wang, T. (2022, May). Transfer attacks revisited: A large-scale empirical study in real computer vision settings. In 2022 IEEE Symposium on Security and Privacy (SP) (pp. 1423-1439). IEEE.

[RKH] Ramirez, M. A., Kim, S. K., Hamadi, H. A., Damiani, E., Byon, Y. J., Kim, T. Y., ... & Yeun, C. Y. (2022). Poisoning attacks and defenses on artificial intelligence: A survey. arXiv preprint arXiv:2202.10276.

[EAV] Eykholt, K., Ahmed, F., Vaishnavi, P., & Rahmati, A. (2024). Taking off the Rose-Tinted Glasses: A Critical Look at Adversarial ML Through the Lens of Evasion Attacks. arXiv preprint arXiv:2410.12076.

[GBA] Guembe, B., Azeta, A., Misra, S., Osamor, V. C., Fernandez-Sanz, L., & Pospelova, V. (2022). The emerging threat of ai-driven cyber attacks: A review. Applied Artificial Intelligence, 36(1), 2037254.

[CJH] Cheng, J., Hussein, M., Billa, J., & AbdAlmageed, W. (2022). Attack-agnostic adversarial detection. arXiv preprint arXiv:2206.00489.

[WYS] Wang, Y., Sun, T., Li, S., Yuan, X., Ni, W., Hossain, E., & Poor, H. V. (2023). Adversarial attacks and defenses in machine learning-empowered communication systems and networks: A contemporary survey. IEEE Communications Surveys & Tutorials.

[SAH] Salem, A. H., Azzam, S. M., Emam, O. E., & Abohany, A. A. (2024). Advancing cybersecurity: a comprehensive review of AI-driven detection techniques. Journal of Big Data, 11(1), 105.

[VAJ] Varma, A. J., Taleb, N., Said, R. A., Ghazal, T. M., Ahmad, M., Alzoubi, H. M., & Alshurideh, M. (2023). A roadmap for smes to adopt an ai based cyber threat intelligence. In The Effect of Information Technology on Business and Marketing Intelligence Systems (pp. 1903-1926). Cham: Springer International Publishing.

[RKZ] Roshan, K., Zafar, A., & Haque, S. B. U. (2024). Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system. Computer Communications, 218, 97-113.

[BTS] Ban, T., Samuel, N., Takahashi, T., & Inoue, D. (2021, August). Combat security alert fatigue with ai-assisted techniques. In Proceedings of the 14th Cyber Security Experimentation and Test Workshop (pp. 9-16).

[AIA] Petihakis, G., Farao, A., Bountakas, P., Sabazioti, A., Polley, J., & Xenakis, C. (2024, July). AIAS: AI-ASsisted cybersecurity platform to defend against adversarial AI attacks. In *Proceedings of the 19th International Conference on Availability, Reliability and Security* (pp. 1-7).

[ZZA] Tian, Z., Cui, L., Liang, J., & Yu, S. (2022). A comprehensive survey on poisoning attacks and countermeasures in machine learning. ACM Computing Surveys, 55(8), 1-35.

[AGS] Agrawal, S. (2022). Enhancing payment security through AI-Driven anomaly detection and predictive analytics. International Journal of Sustainable Infrastructure for Cities and Societies, 7(2), 1-14.

[TZC] Tian, Z., Cui, L., Liang, J., & Yu, S. (2022). A comprehensive survey on poisoning attacks and countermeasures in machine learning. ACM Computing Surveys, 55(8), 1-35.

[HRK] Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. Journal of Big Data, 11(1), 113.

[NAB] Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. Ieee Access, 9, 78658-78700.

[TRF] Tramer, F. (2022, June). Detecting adversarial examples is (nearly) as hard as classifying them. In International Conference on Machine Learning (pp. 21692-21702). PMLR.

[ABA] Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99% false positives: A qualitative study of {SOC} analysts' perspectives on security alarms. In 31st USENIX Security Symposium (USENIX Security 22) (pp. 2783-2800).

[DMS] Daoud, M. S., Shehab, M., Al-Mimi, H. M., Abualigah, L., Zitar, R. A., & Shambour, M. K. Y. (2023). Gradient-based optimizer (gbo): a review, theory, variants, and applications. Archives of Computational Methods in Engineering, 30(4), 2431-2449.

[IMK] Ivanovs, M., Kadikis, R., & Ozols, K. (2021). Perturbation-based methods for explaining deep neural networks: A survey. Pattern Recognition Letters, 150, 228-234.

[LYG] Li, Y., Guo, L., Liu, Y., Liu, J., & Meng, F. (2021). A temporal-spectral-based squeeze-and-excitation feature fusion network for motor imagery EEG decoding. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 29, 1534-1545.

[GAH] Guesmi, A., Hanif, M. A., Ouni, B., & Shafique, M. (2023). Physical adversarial attacks for camera-based smart systems: Current trends, categorization, applications, research challenges, and future outlook. IEEE Access.

[HOT] Hossain, T. (2022). A Comparative Analysis of Adversarial Capabilities, Attacks, and Defenses Across the Machine Learning Pipeline in White-Box and Black-Box Settings. Applied Research in Artificial Intelligence and Cloud Computing, 5(1), 195-212.

[PSR] Parisineni, S. R. A., & Pal, M. (2024). Enhancing trust and interpretability of complex machine learning models using local interpretable model agnostic shap explanations. International Journal of Data Science and Analytics, 18(4), 457-466.

[NYM] Nohara, Y., Matsumoto, K., Soejima, H., & Nakashima, N. (2022). Explanation of machine learning models using shapley additive explanation and application for real data in hospital. Computer Methods and Programs in Biomedicine, 214, 106584.

[JNT] Jeffrey, N., Tan, Q., & Villar, J. R. (2024). A hybrid methodology for anomaly detection in Cyber–Physical Systems. Neurocomputing, 568, 127068.

[SIH] Sarker, I. H. (2023). Multi-aspects AI-based modeling and adversarial learning for cybersecurity intelligence and robustness: A comprehensive overview. Security and Privacy, 6(5), e295.

[IND] Ilg, N., Duplys, P., Sisejkovic, D., & Menth, M. (2023). Survey of contemporary open-source honeypots, frameworks, and tools. Journal of Network and Computer Applications, 103737.

[SAK] Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., & Calinescu, A. (2022). Digital twins: State of the art theory and practice, challenges, and open research questions. Journal of Industrial Information Integration, 30, 100383.

[KOV] Kostyumov, V. (2022). A survey and systematization of evasion attacks in computer vision. International Journal of Open Information Technologies, 10(10), 11-20.

[TZC] Tian, Z., Cui, L., Liang, J., & Yu, S. (2022). A comprehensive survey on poisoning attacks and countermeasures in machine learning. ACM Computing Surveys, 55(8), 1-35.

[WFN] Waseda, F., Nishikawa, S., Le, T. N., Nguyen, H. H., & Echizen, I. (2023). Closer look at the transferability of adversarial examples: How they fool different models differently. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1360-1368).

[FHQ] Fang, H., Qiu, Y., Yu, H., Yu, W., Kong, J., Chong, B., ... & Xu, K. (2024). Privacy leakage on dnns: A survey of model inversion attacks and defenses. arXiv preprint arXiv:2402.04013.

[CAA] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., & Mukhopadhyay, D. (2021). A survey on adversarial attacks and defences. CAAI Transactions on Intelligence Technology, 6(1), 25-45.

[SAF] Mahboubi, A., Luong, K., Aboutorab, H., Bui, H. T., Jarrad, G., Bahutair, M., ... & Gately, H. (2024). Evolving techniques in cyber threat hunting: A systematic review. Journal of Network and Computer Applications, 104004.

[ATA] Ahanger, T. A., Aljumah, A., & Atiquzzaman, M. (2022). State-of-the-art survey of artificial intelligent techniques for IoT security. Computer Networks, 206, 108771.

[CAF] Cooper, A. F., Levy, K., & De Sa, C. (2021, October). Accuracy-Efficiency Trade-Offs and Accountability in Distributed ML Systems. In Proceedings of the 1st ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (pp. 1-11).

[OLJ] Oldemeyer, L., Jede, A., & Teuteberg, F. (2024). Investigation of artificial intelligence in SMEs: a systematic review of the state of the art and the main implementation challenges. Management Review Quarterly, 1-43.

[AKM] Amekoe, K. M., Azzag, H., Dagdia, Z. C., Lebbah, M., & Jaffre, G. (2024). Exploring accuracy and interpretability trade-off in tabular learning with novel attention-based models. Neural Computing and Applications, 36(30), 18583-18611.

[AAS] Ami, A. S., Moran, K., Poshyvanyk, D., & Nadkarni, A. (2024, May). " False negative-that one is going to kill you": Understanding Industry Perspectives of Static Analysis based Security Testing. In 2024 IEEE Symposium on Security and Privacy (SP) (pp. 3979-3997). IEEE.

[EHM] El-Hajj, M. (2024). Leveraging Digital Twins and Intrusion Detection Systems for Enhanced Security in IoT-Based Smart City Infrastructures. Electronics, 13(19), 3941.

[ZLT] Zhang, L., & Thing, V. L. (2021). Three decades of deception techniques in active cyber defense-retrospect and outlook. Computers & Security, 106, 102288.

[CBJ] Chander, B., John, C., Warrier, L., & Gopalakrishnan, K. (2024). Toward trustworthy artificial intelligence (TAI) in the context of explainability and robustness. ACM Computing Surveys.

[BSS] Balantrapu, Siva Subrahmanyam. "Adversarial Machine Learning: Security Threats and Mitigations." International Journal of Sustainable Development in Computing Science 1.3 (2019): 1-18.

[QSR] Qiu, Shilin, et al. "Review of artificial intelligence adversarial attack and defense technologies." Applied Sciences 9.5 (2019): 909.

[HHB] Hosseini, Hossein, et al. "Blocking transferability of adversarial examples in black-box learning systems." arXiv preprint arXiv:1703.04318 (2017).

[SSG] Salah, Saeed, Gabriel Maciá-Fernández, and Jesús E. Díaz-Verdejo. "A model-based survey of alert correlation techniques." Computer Networks 57.5 (2013): 1289-1317.

[HKD] He, Ke, Dan Dongseong Kim, and Muhammad Rizwan Asghar. "Adversarial machine learning for network intrusion detection systems: A comprehensive survey." IEEE Communications Surveys & Tutorials 25.1 (2023): 538-566.

[CAA] Chakraborty, Anirban, et al. "Adversarial attacks and defences: A survey." arXiv preprint arXiv:1810.00069 (2018).

[TKA] Tam, K., et al. "Adversarial AI Testcases for Maritime Autonomous Systems." (2023).

[ARD] Alford, Ron, Dean Lawrence, and Michael Kouremetis. "Caldera: A red-blue cyber operations automation platform." MITRE: Bedford, MA, USA (2022).

[ARO] Atomic Red Team framework [1] https://www.atomicredteam.io/ [Online] [Last access 11/12/2024]

[CAL] Caldera https://caldera.mitre.org/ [Online] [Last access 11/12/2024]

[MAT] Mitre att&ck, https://attack.mitre.org/ [Online] [Last access 11/12/2024]

[CIS] Cesnt Center for Internet Security, https://www.cisecurity.org/ [Online] [Last access 11/12/2024]

[SCF] Secure Controls Framework, https://securecontrolsframework.com/ [Online] [Last access 11/12/2024]

[PVF] Pantelakis, Vasileios, et al. "Adversarial machine learning attacks on multiclass classification of iot network traffic." Proceedings of the 18th International Conference on Availability, Reliability and Security. 2023.

[GEO] Georgescu, T.M. and Smeureanu, I., 2017. Using ontologies in cybersecurity field. Informatica Economica, 21(3), p.5

[AMN] Al-Mohannadi, H., Mirza, Q., Namanya, A., Awan, I., Cullen, A. and Disso, J., 2016, August. Cyber-attack modeling analysis techniques: An overview. In 2016 IEEE 4th international conference on future internet of things and cloud workshops (FiCloudW) (pp. 69-76). IEEE.

[OKU] Okutan, A., Werner, G., Yang, S.J. and McConky, K., 2018. Forecasting cyberattacks with incomplete, imbalanced, and insignificant data. Cybersecurity, 1, pp.1-16.

[SHK] Stojanović, B., Hofer-Schmitz, K. and Kleb, U., 2020. APT datasets and attack modeling for automated detection methods: A review. Computers & Security, 92, p.101734.

[UHH] Uetz, R., Hemminghaus, C., Hackländer, L., Schlipper, P. and Henze, M., 2021, December. Reproducible and adaptable log data generation for sound cybersecurity experiments. In Proceedings of the 37th Annual Computer Security Applications Conference (pp. 690-705).

[SOC] SOCBED https://github.com/fkie-cad/socbed [Online] [Last access 11/12/2024]

[STE] Steverson, K., Carlin, C., Mullin, J. and Ahiskali, M., 2021, May. Cyber intrusion detection using natural language processing on windows event logs. In 2021 International Conference on Military Communication and Information Systems (ICMCIS) (pp. 1-7). IEEE.

[AHM] Ahmetoglu, H. and Das, R., 2022. A comprehensive review on detection of cyber-attacks: Data sets, methods, challenges, and future research directions. Internet of Things, 20, p.100615.

[ALJ] Aljabri, M., Alahmadi, A.A., Mohammad, R.M.A., Aboulnour, M., Alomari, D.M. and Almotiri, S.H., 2022. Classification of firewall log data using multiclass machine learning models. Electronics, 11(12), p.1851.

[LSF] Landauer, M., Skopik, F., Frank, M., Hotwagner, W., Wurzenberger, M. and Rauber, A., 2022. Maintainable log datasets for evaluation of intrusion detection systems. IEEE Transactions on Dependable and Secure Computing, 20(4), pp.3466-3482.

[RAD] Karapoola, S., Singh, N., Rebeiro, C. and V, K., 2022, October. RaDaR: A Real-Word Dataset for AI powered Run-time Detection of Cyber-Attacks. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (pp. 3222-3232).

[YCL] Yang, C.T., Chan, Y.W., Liu, J.C., Kristiani, E. and Lai, C.H., 2022. Cyberattacks detection and analysis in a network log system using XGBoost with ELK stack. Soft Computing, 26(11), pp.5143-5157.

[NFL] NetFlow https://www.cisco.com/c/es_mx/tech/quality-of-service-qos/netflow/index.html [Online] [Last access 11/12/2024]

[DTP] Shandilya, S.K., Ganguli, C., Izonin, I. and Nagar, A.K., 2023. Cyber attack evaluation dataset for deep packet inspection and analysis. Data in Brief, 46, p.108771.

[DPD] Shishir Kumar Shandilya, Chirag Ganguli, Ivan Izonin, Atulya Kumar Nagar, September 24, 2022, "Cyber Attack Evaluation Dataset for Deep Packet Inspection and Analysis", IEEE Dataport, doi: https://dx.doi.org/10.21227/a8m1-x573.

[TUS] Tushkanova, O., Levshun, D., Branitskiy, A., Fedorchenko, E., Novikova, E. and Kotenko, I., 2023. Detection of cyberattacks and anomalies in cyber-physical systems: Approaches, data sources, evaluation. Algorithms, 16(2), p.85.

[SYJ] Shih, W.C., Yang, C.T., Jiang, C.T. and Kristiani, E., 2023. Implementation and visualization of a netflow log data lake system for cyberattack detection using distributed deep learning. The Journal of Supercomputing, 79(5), pp.4983-5012.

[PRA] Prasad, A. and Chandra, S., 2023. Machine learning to combat cyberattack: a survey of datasets and challenges. The Journal of Defense Modeling and Simulation, 20(4), pp.577-588.

[KOK] Kovačević, I., Komadina, A., Štengl, B. and Groš, S., 2023, May. Light-weight synthesis of security logs for evaluation of anomaly detection and security related experiments. In Proceedings of the 16th European Workshop on System Security (pp. 30-36).

[IRS] Irshad, E. and Siddiqui, A.B., 2023. Cyber threat attribution using unstructured reports in cyber threat intelligence. Egyptian Informatics Journal, 24(1), pp.43-59.

[OSI] 15 top open-source intelligence tools https://www.csoonline.com/article/567859/what-is-osint-top-open-source-intelligence-tools.html [Online] [Last access 11/12/2024]

[KCT] Koloveas, P., Chantzios, T., Tryfonopoulos, C. and Skiadopoulos, S., 2019, July. A crawler architecture for harvesting the clear, social, and dark web for IoT-related cyber-threat intelligence. In 2019 IEEE World Congress on Services (SERVICES) (Vol. 2642, pp. 3-8). IEEE.

[KIM] Kim, H., Kim, I. and Kim, K., 2021. AIBFT: artificial intelligence browser forensic toolkit. Forensic Science International: Digital Investigation, 36, p.301091.

[ASH] Alshammery, M.K. and Aljuboori, A.F., 2022. Crawling and mining the dark web: A survey on existing and new approaches. Iraqi Journal of Science, pp.1339-1348

[WEE] Weerasinghe, M., Maduranga, M.W.P. and Kawya, M.V.T., 2023. Enhancing Web Scraping with Artificial Intelligence: A Review.

[BPO] Bergman, J. and Popov, O.B., 2023. Exploring dark web crawlers: a systematic literature review of dark web crawlers and their implementation. IEEE Access, 11, pp.35914-35933.

[WAN] Wan, B., Xu, C. and Koo, J., 2023. Exploring the Effectiveness of Web Crawlers in Detecting Security Vulnerabilities in Computer Software Applications. International Journal of Informatics and Information Systems, 6(2), pp.56-65

[IBC] Bergman, J. and Popov, O.B., 2023. Exploring dark web crawlers: a systematic literature review of dark web crawlers and their implementation. IEEE Access, 11, pp.35914-35933

[KHD] Khder, M.A., 2021. Web scraping or web crawling: State of art, techniques, approaches and application. International Journal of Advances in Soft Computing & Its Applications, 13(3).

[EPI] Epiphaniou, G., French, T. and Maple, C., 2014. The dark Web: Cyber-security intelligence gathering opportunities, risks and rewards. Journal of computing and information technology, 22(LISS 2013), pp.21-30

[SCF] Schäfer, M., Fuchs, M., Strohmeier, M., Engel, M., Liechti, M. and Lenders, V., 2019, May. BlackWidow: Monitoring the dark web for cyber security information. In 2019 11th International Conference on Cyber Conflict (CyCon) (Vol. 900, pp. 1-21). IEEE

[ARN] Arnold, N., Ebrahimi, M., Zhang, N., Lazarine, B., Patton, M., Chen, H. and Samtani, S., 2019, July. Dark-net ecosystem cyber-threat intelligence (CTI) tool. In 2019 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 92-97). IEEE.

[BAS] Basheer, R. and Alkhatib, B., 2021. Threats from the dark: a review over dark web investigation research for cyber threat intelligence. Journal of Computer Networks and Communications, 2021(1), p.1302999

[SCH] Samtani, S., Chai, Y. and Chen, H., 2022. LINKING EXPLOITS FROM THE DARK WEB TO KNOWN VULNERABILITIES FOR PROACTIVE CYBER THREAT INTELLIGENCE: AN ATTENTION-BASED DEEP STRUCTURED SEMANTIC MODEL1. MIS quarterly, 46(2).

[SAP] Sapienza, A., Ernala, S.K., Bessi, A., Lerman, K. and Ferrara, E., 2018, April. Discover: Mining online chatter for emerging cyber threats. In Companion Proceedings of the The Web Conference 2018 (pp. 983-990).

[WSP] Williams, R., Samtani, S., Patton, M. and Chen, H., 2018, November. Incremental hacker forum exploit collection and classification for proactive cyber threat intelligence: An exploratory study. In 2018 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 94-99). IEEE.

[TAK] Takahashi, T., Panta, B., Kadobayashi, Y. and Nakao, K., 2018. Web of cybersecurity: Linking, locating, and discovering structured cybersecurity information. International Journal of Communication Systems, 31(3), p.e3470

[KRI] Krishnan, N. and Deepak, G., 2021, March. KnowCrawler: AI classification cloud-driven framework for web crawling using collective knowledge. In European, Asian, Middle Eastern, North African Conference on Management & Information Systems (pp. 371-382). Cham: Springer International Publishing

[VLA] Vlachos, V., Stamatiou, Y.C., Tzamalis, P. and Nikoletseas, S., 2022. The SAINT observatory subsystem: an open-source intelligence tool for uncovering cybersecurity threats. International Journal of Information Security, 21(5), pp.1091-1106.

[BLA] Blasch, E., Al-Nashif, Y. and Hariri, S., 2014. Static versus dynamic data information fusion analysis using DDDAS for cyber security trust. Procedia Computer Science, 29, pp.1299-1313.

[ALS] Alsmadi, I.M., Karabatis, G. and Aleroud, A. eds., 2017. Information fusion for cyber-security analytics (Vol. 691). Switzerland: Springer International Publishing.

[FAT] Fatima, H., Satpathy, S., Mahapatra, S., Dash, G.N. and Pradhan, S.K., 2017, March. Data fusion & visualization application for network forensic investigation-a case study. In 2017 2nd International Conference on Anti-Cyber Crimes (ICACC) (pp. 252-256). IEEE.

[KEN] Kennedy, M., 2018. Data Fusion of Security Logs to Measure Critical Security Controls to Increase Situation Awareness.

[COS] Costa, P.C., Yu, B., Atiahetchi, M. and Myers, D., 2018, July. High-level information fusion of cyber-security expert knowledge and experimental data. In 2018 21st International Conference on Information Fusion (FUSION) (pp. 2322-2329). IEEE.

[JUG] Ju, A., Guo, Y., Ye, Z., Li, T. and Ma, J., 2019. Hetemsd: A big data analytics framework for targeted cyber-attacks detection using heterogeneous multisource data. Security and Communication Networks, 2019(1), p.5483918.

[BOH] Bohara, A., 2020. Information-fusion-based methods to improve the detection of advanced cyber threats (Doctoral dissertation, University of Illinois at Urbana-Champaign).

[YUM] Yu, S., Mueller, P. and Qian, J. eds., 2020. Security and Privacy in Digital Economy: First International Conference, SPDE 2020, Quzhou, China, October 30–November 1, 2020, Proceedings (Vol. 1268). Springer Nature.

[DIW] Diwan, T.D., 2021. An investigation and analysis of cyber security information systems: latest trends and future suggestion. Information Technology in Industry, 9(2), pp.477-492.

[KOL] Koloveas, P., Chantzios, T., Alevizopoulou, S., Skiadopoulos, S. and Tryfonopoulos, C., 2021. intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence. Electronics, 10(7), p.818.

[ANJ] Anjum, N., Latif, Z., Lee, C., Shoukat, I.A. and Iqbal, U., 2021. Mind: A multi-source data fusion scheme for intrusion detection in networks. Sensors, 21(14), p.4941.

[CHE] Chen, Z., 2022. Observations and expectations on recent developments of data lakes. Procedia Computer Science, 214, pp.405-411.

[HUS] Hussen, N., Elghamrawy, S.M., Salem, M. and El-Desouky, A.I., 2023. A fully streaming big data framework for cyber security based on optimized deep learning algorithm. IEEE Access, 11, pp.65675-65688.

[GUO] Guo, Y., Liu, Z., Huang, C., Wang, N., Min, H., Guo, W. and Liu, J., 2023. A framework for threat intelligence extraction and fusion. Computers & Security, 132, p.103371.

[ZHU] Zhu, D., Yin, H., Xu, Y., Wu, J., Zhang, B., Cheng, Y., Yin, Z., Yu, Z., Wen, H. and Li, B., 2023. A survey of advanced information fusion system: From model-driven to knowledge-enabled. Data Science and Engineering, 8(2), pp.85-97.

[HED] He, X., Dong, H., Yang, W. and Li, W., 2023. Multi-Source Information Fusion Technology and Its Application in Smart Distribution Power System. Sustainability, 15(7), p.6170.

[YAK] Yadav, A., Kumar, A. and Singh, V., 2023. Open-source intelligence: a comprehensive review of the current state, applications and future perspectives in cyber security. Artificial Intelligence Review, 56(11), pp.12407-12438.

[JEM] Jemili, F., 2023. Towards data fusion-based big data analytics for intrusion detection. Journal of Information and Telecommunication, 7(4), pp.409-436.

[ABI] Abid, A., Jemili, F. and Korbaa, O., 2024. Real-time data fusion for intrusion detection in industrial control systems based on cloud computing and big data techniques. Cluster Computing, 27(2), pp.2217-2238.

[NYA] Nyaletey, E., Parizi, R.M., Zhang, Q. and Choo, K.K.R., 2019, July. BlockIPFS-blockchain-enabled interplanetary file system for forensic and trusted data traceability. In 2019 IEEE International Conference on Blockchain (Blockchain) (pp. 18-25). IEEE.

[MUK] Mukne, H., Pai, P., Raut, S. and Ambawade, D., 2019, July. Land record management using hyperledger fabric and ipfs. In 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-8). IEEE.

[HAN] Hanafi, J., Prayudi, Y. and Luthfi, A., 2021. Ipfschain: Interplanetary file system and hyperledger fabric collaboration for chain of custody and digital evidence management. International Journal of Computer Applications, 183(41), pp.24-32.

[MAN] Mani, V., Manickam, P., Alotaibi, Y., Alghamdi, S. and Khalaf, O.I., 2021. Hyperledger healthchain: patient-centric IPFS-based storage of health records. Electronics, 10(23), p.3003.

[PIL] Pilares, I.C.A., Azam, S., Akbulut, S., Jonkman, M. and Shanmugam, B., 2022. Addressing the challenges of electronic health records using blockchain and ipfs. Sensors, 22(11), p.4032.

[ZWZ] Zhao, X., Wang, S., Zhang, Y. and Wang, Y., 2022. Attribute-based access control scheme for data sharing on hyperledger fabric. Journal of Information Security and Applications, 67, p.103182.

[PIN] Pingos, M., Christodoulou, P. and Andreou, A., 2022, July. DLMetaChain: An IoT data lake architecture based on the blockchain. In 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA) (pp. 1-8). IEEE.

[CZY] Chen, J., Zhang, C., Yan, Y. and Liu, Y., 2022. FileWallet: A File Management System Based on IPFS and Hyperledger Fabric. CMES-Computer Modeling in Engineering & Sciences, 130(2).

[CYT] Chen, C.L., Yang, J., Tsaur, W.J., Weng, W., Wu, C.M. and Wei, X., 2022. Enterprise data sharing with privacy-preserved based on hyperledger fabric blockchain in IIOT’s application. Sensors, 22(3), p.1146.

[AAJ] Ali, H., Ahmad, J., Jaroucheh, Z., Papadopoulos, P., Pitropakis, N., Lo, O., Abramson, W. and Buchanan, W.J., 2022. Trusted threat intelligence sharing in practice and performance benchmarking through the hyperledger fabric platform. Entropy, 24(10), p.1379.

[MPV] Machado, J.M., Prieto, J., Vieira, P., Peixoto, H., Abelha, A., Arroyo, D. and Vigneri, L., 2023. Blockchain and Applications, 5th International Congress. Springer.

[KVH] Khaleelullah, S., Vangapalli, S.T., Gaddam, M., Hanumakonda, V.S. and Gangapuram, U.K.G., 2023, June. Verification of academic records using hyperledger fabric and ipfs. In 2023 3rd International Conference on Pervasive Computing and Social Networking (ICPCSN) (pp. 210-217). IEEE.

[HAR] Hariyanto, F. and Ramli, K., 2024. Design and Analysis of Cybersecurity Information Sharing Mechanism Between Computer Security Incident Response Teams (CSIRT) in Indonesia on Blockchain Technology Through Hyperledger Composer and Interplanetary File System (IPFS). MALCOM: Indonesian Journal of Machine Learning and Computer Science, 4(4), pp.1390-1402.

[WWQ] Wen, F., Wang, Z., Qu, L., Huang, H. and Hu, X., 2024. Enhancing secure multi-group data sharing through integration of IPFS and hyperledger fabric. PeerJ Computer Science, 10, p.e1962.

[MAW] Traffic Trace Page https://mawi.wide.ad.jp/mawi/samplepoint-F/2024/ [Online] [Last access 11/12/2024]

[GPD] Gramegna, Alex, and Paolo Giudici. "SHAP and LIME: an evaluation of discriminative power in credit risk." Frontiers in Artificial Intelligence 4 (2021): 752558.

[WNC] Kaspersky, What is WannaCry ransomware? https://www.kaspersky.com/resource-center/threats/ransomware-wannacry [Online] [Last access 11/12/2024]

[NTY] Vinciworks, NotPetya: The World’s Worst Cyber Attack, https://vinciworks.com/blog/notpetya-the-worlds-worst-cyber-attack/ [Online] [Last access 11/12/2024]

[LPF] Leonidou, Pantelitsa, et al. "A qualitative analysis of illicit arms trafficking on darknet marketplaces." Proceedings of the 18th International Conference on Availability, Reliability and Security. 2023.

[DIDS] 1998 DARPA Intrusion Detection Evaluation Dataset, https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset [Online] [Last access 11/12/2024]

[KDDC] KDD Cup 1999 Data, https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Online] [Last access 11/12/2024]

[KSLK] NSL-KDD dataset 2009, https://www.impactcybertrust.org/dataset_view?idDataset=928 [Online] [Last access 11/12/2024]

[UNSWN] The UNSW-NB15 Dataset https://research.unsw.edu.au/projects/unsw-nb15-dataset [Online] [Last access 11/12/2024]

[CCIDS] CIC-IDS-Collection https://www.kaggle.com/datasets/dhoogla/cicidscollection [Online] [Last access 11/12/2024]

[VAF] Voudouris, Anastassios, et al. "Integrating Hyperledger Fabric with Satellite Communications: A Revolutionary Approach for Enhanced Security and Decentralization in

Space Networks." Proceedings of the 19th International Conference on Availability, Reliability and Security. 2024.