electronics

MDPI

Article

Utilisation of Artificial Intelligence and Cybersecurity Capabilities: A Symbiotic Relationship for Enhanced Security and Applicability

Ed Kamya Kiyemba Edris

School of Physics, Engineering and Computer Science, University of Hertfordshire, Computer Science 1, Hatfield AL10 9EU, UK; e.edris@herts.ac.uk

Abstract: The increasing interconnectivity between physical and cyber-systems has led to more vulnerabilities and cyberattacks. Traditional preventive and detective measures are no longer adequate to defend against adversaries. Artificial Intelligence (AI) is used to solve complex problems, including those of cybersecurity. Adversaries also utilise AI for sophisticated and stealth attacks. This study aims to address this problem by exploring the symbiotic relationship of AI and cybersecurity to develop a new, adaptive strategic approach to defend against cyberattacks and improve global security. This paper explores different disciplines to solve security problems in real-world contexts, such as the challenges of scalability and speed in threat detection. It develops an algorithm and a detective predictive model for a Malicious Alert Detection System (MADS) that is an integration of adaptive learning and a neighbourhood-based voting alert detection framework. It evaluates the model's performance and efficiency among different machines. The paper discusses Machine Learning (ML) and Deep Learning (DL) techniques, their applicability in cybersecurity, and the limitations of using AI. Additionally, it discusses issues, risks, vulnerabilities, and attacks against AI systems. It concludes by providing recommendations on security for AI and AI for security, paving the way for future research on enhancing AI-based systems and mitigating their risks.

Keywords: artificial intelligence; cybersecurity; machine learning; detection; alerts; adversarial attacks; predictive model; evaluation

Academic Editor: Aryya Gangopadhyay

Received: 15 March 2025
Revised: 11 May 2025
Accepted: 16 May 2025
Published: 19 May 2025

Citation: Edris, E.K.K. Utilisation of Artificial Intelligence and Cybersecurity Capabilities: A Symbiotic Relationship for Enhanced Security and Applicability. *Electronics* **2025**, *14*, 2057. https://doi.org/10.3390/electronics14102057

1. Introduction

Cybersecurity is a fast-evolving discipline, and threat actors (TAs) constantly endeavour to stay ahead of security teams with new and sophisticated attacks. The use of interconnected devices to access data ubiquitously has increased exponentially, raising more security concerns. Traditional security solutions are becoming inadequate in detecting and preventing such attacks. However, advances in cryptographic and Artificial Intelligence (AI) techniques show promise in enabling cybersecurity experts to counter such attacks [1]. AI is being leveraged to solve a number of problems, from using chatbots to virtual assistants to automation, allowing humans to focus on higher-value work; they are also used for predictions, analytics, and cybersecurity.

Recovery from a data breach costs USD 4.35 million on average and takes 196 days [2]. Organisations are increasingly investing in cybersecurity, adding AI enablement to improve threat detection, incident response (IR), and compliance. Patterns in data can be recognised using Machine Learning (ML), monitoring, and threat intelligence to enable systems to learn from past events. It is estimated that AI in the cybersecurity market will be worth

USD 102.78 billion by 2032 globally [3]. Currently, different initiatives are defining new standards and certifications to elicit users' trust in AI. The adoption of AI can improve best practices and security posture, but it can also create new forms of attacks. Therefore, secure, trusted, and reliable AI is necessary, achieved by integrating security in the design, development, deployment, operation, and maintenance stages.

Failures of AI systems are becoming common, and it is crucial to understand them and prevent them as they occur [4,5]. Failures in general AI have a higher impact than in narrow AI, but a single failure in a superintelligent system might result in a catastrophic disaster with no prospect of recovery. AI safety can be improved with cybersecurity practices, standards, and regulations. Its risks and limitations should also be understood, and solutions should be developed. Systems commonly experience recurrent problems with scalability, accountability, context, accuracy, and speed in the field of cybersecurity [6]. The inventory of ML algorithms, techniques, and systems have to be explored through the lens of security. Considering the fact that AI can fail, there should be models in place to make AI decisions explainable [7].

How can an ML-driven system effectively detect and predict cybersecurity incidents in a dynamic and scalable multimachine environment using alert-level intelligence? This paper addresses this question by exploring AI and cybersecurity with use cases and practical concepts in a real-world context based on the field experience of the authors. It gives an overview of how AI and cybersecurity empower each other symbiotically. It discusses the main disciplines of AI and how they can be applied to solve complex cybersecurity problems and the challenges of data analytics. It highlights the need to evaluate both AI for security and security for AI to deploy safe, trusted, secure AI-driven applications. Furthermore, it develops an algorithm and a predictive model, the Malicious Alert Detection System (MADS), to demonstrate AI applicability. It evaluates the model's performance; proposes methods to address AI-related risks, limitations, and attacks; explores evaluation techniques; and recommends a safe and successful adoption of AI.

The rest of this paper is structured as follows. Section 2 reviews related work on AI and cybersecurity. In Section 3, the security objectives and AI foundational concepts are presented. Section 4 discusses how AI can be leveraged to solve security problems. The utilisation of ML algorithms is presented in Section 6. Section 5 presents a case study of a predictive AI model, MADS, and its evaluation. The risks and limitations of AI are discussed in Section 7. This paper is concluded with remarks and a discussion of future works in Section 8.

2. Background

Security breaches and loss of confidential data are still big challenges for organisations. With the increased sophistication of modern attacks, there is a need to detect malicious activities and also to predict the steps that will be taken by an adversary. This can be achieved through the utilisation of AI by applying it to use cases such as traffic monitoring, authentication, and anomalous behaviour [8].

Current AI research involves search algorithms, knowledge graphs, Natural Languages Processing (NLP), expert systems, ML, and Deep Learning (DL), while the development process includes perceptual, cognitive, and decision-making intelligence. The integration of cybersecurity with AI has huge benefits, such as improving the efficiency and performance of security systems and providing better protection from cyber-threats. It can improve an organisation's security maturity by adopting a holistic view, combining AI with human insight. Thus, the socially responsible use of AI is essential to further mitigate related concerns [9].

The speed of processes and the amount of data used in defending cyberspace cannot be handled without automation. AI techniques are being introduced to construct smart models for malware classification, intrusion detection, and threat intelligence gathering [10]. Nowadays, it is difficult to develop software with conventional fixed algorithms to defend against dynamically evolving cyberattacks [11]. AI can provide flexibility and learning capability to software development. However, TAs have also figured out how to exploit AI and use it to carry out new attacks.

Moreover, ML and neural network (NN) policies in Reinforcement Learning (RL) methods are vulnerable to adversarial learning attacks, which aim at decreasing the effectiveness of threat detection [12–14]. AI models face threats that disturb their data, learning, and decision-making. Deep Reinforcement Learning (DRL) can be utilised to optimise security defence against adversaries using proactive and adaptive countermeasures [15]. The use of a recursive algorithm, DL, and inference in NNs have enabled inherent advantages over existing computing frameworks [16]. AI-enabled applications can be combined with human emotions, cognitions, social norms, and behavioural responses [17] to improve societal issues. However, the use of AI can lead to ethical and legal issues, which are already big problems in cybersecurity. There are significant concerns about data privacy and applications' transparency. It is also important to address the criminal justice issues related to AI usage, liability, and damage compensation.

Recent advancements introduced DL techniques, such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, which have demonstrated improved detection capabilities in temporal and structured data streams [18,19]. Despite their accuracy, DL models are often considered "black-box" in nature, providing little transparency or control to cybersecurity analysts [20]. Moreover, these models typically classify individual events in isolation, lacking the contextual correlation necessary for understanding attack campaigns or incidents. Studies show that AI-based detectors can be deceived through carefully crafted inputs [21]. This vulnerability underscores the need for robust and interpretable detection architectures that not only predict threats but also do so with resilience to adversarial manipulation.

In light of these limitations, MADS distinguishes itself by integrating ML prediction, k-nearest neighbours (k-NN)-based voting [22], and incident-level correlation within a sliding window framework. Unlike conventional ML classifiers that operate in a stateless manner, it retains a temporal alert history, allowing for contextual decisions based on cumulative patterns [23]. Its hybrid decision-making process reduces susceptibility to isolated misclassifications and enhances trust through interpretable thresholds and voting confidence. Furthermore, MADS is explicitly designed to handle multisource alert streams from distributed machines, a capability often overlooked in single-node systems.

The existing AI-based solutions focus on enhancing detection accuracy or reducing false alarms [24], but few address holistic incident formation and operational scalability across networked devices. This paper bridges this gap by introducing a multistage detection and aggregation pipeline that is both adaptive and transparent, making it well suited for modern threat environments. It introduces a novel integration of DL and neighbourhood-based voting within a multimachine alert stream processing framework.

2.1. Artificial Intelligence

For a system to be considered to have AI capability, it must have at least one of the six foundational capabilities to pass the Turing Test [25] and the Total Turing Test [26]. These give AI the ability to understand the natural language of a human being, store and process information, reason, learn from new information, see and perceive objects in the

environment, and manipulate and move physical objects. Some advanced AI agents may possess all six capabilities.

The advancements of AI will accelerate, making it more complex and ubiquitous, leading to the creation of a new level of AI. There are three levels of AI [27]:

1. Artificial Narrow Intelligence (ANI): The first level of AI that specialises in one area but cannot solve problems in other areas autonomously.
2. Artificial General Intelligence (AGI): AI that reaches the intelligence level of a human, having the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly, and learn from experience.
3. Artificial Super Intelligence (ASI): AI that is far superior to the best human brain in all cognitive domains, such as creativity, knowledge, and social abilities.

Even ANI models have been able to disrupt technology unexpectedly, such as generative and agentic AI, chatbots, and predictive models. AI enables security systems such as Endpoint Detection and Response (EDR) and IDS to store, process, and learn from huge amounts of data [28,29]. These data are ingested from network devices, workstations, and the Application Programming Interface (API), which is used to identify patterns such as sign-in logs, locations, time-zone, connection types, and abnormal behaviours. These applications can take actions using the associated sign-in risk score and security policies to automatically block login attempts or enforce strong authentication requirements [30].

2.1.1. Learning and Decision-Making

The discipline of learning is the foundation of AI. The ability to learn from input data moves systems away from the rule-based programming approach. An AI-enabled malware detection system operates differently from a traditional signature-based system. Rather than relying on a predefined list of virus signatures, the system is trained using data to identify abnormal program execution patterns known as behaviour-based or anomaly detection, which is a foundational technique used in malware and intrusion detection systems [31].

In the traditional approach, a software engineer identifies all possible inputs and conditions that software will be subjected to, but if the program receives an input that it is not designed to handle, it fails [32]. For instance, when searching for a Structured Query Language (SQL) injection in server logs, in the programming approach, the vulnerability scanner will continuously look for parameters that are not within limits [33]. Additionally, it is complicated to manage multiple vulnerabilities with traditional vulnerability management methods. However, an intelligent vulnerability scanner foresees the possible combinations and ranges, using a learning-based approach. The training data, like source codes or program execution contexts, are fed to the model to learn and act on new data following the ML pipeline in Figure 1 [34].

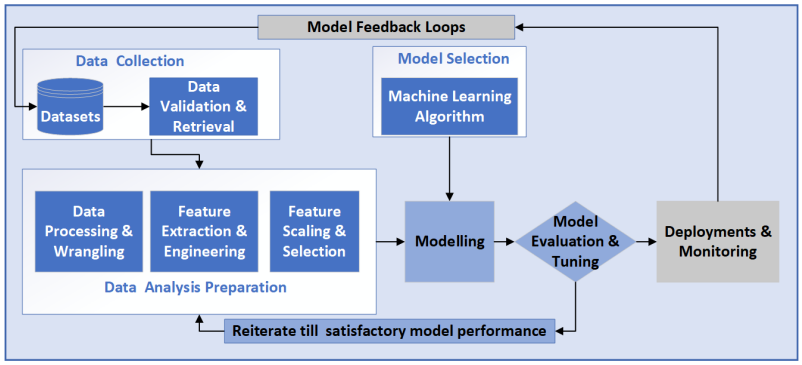


Figure 1. ML pipeline.

2.1.2. Artificial Intelligence and Cybersecurity

An intelligent agent is used to maximise the probability of goal completion. It is fed huge amounts of data, learns patterns, analyses new data, and presents it with recommendations for analysts to make decisions, as shown in Figure 2 [35]. AI can be used to complement traditional tools together with policies, processes, personnel, and methods to minimise security breaches.

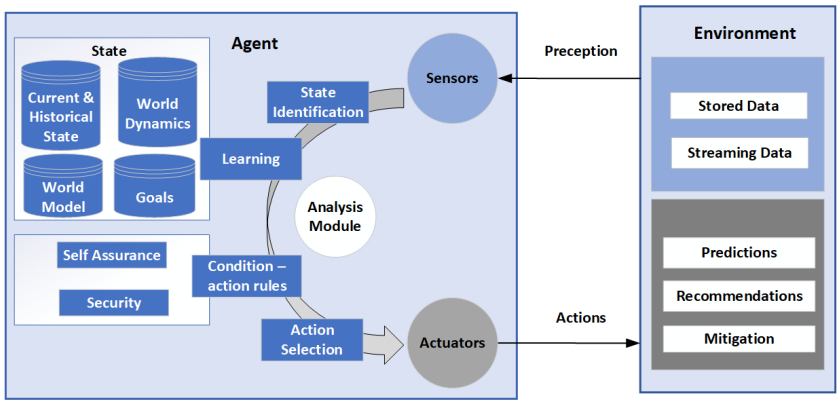


Figure 2. AI agent.

Utilising AI can improve the efficiency of vulnerability assessment with better accuracy and make sense of statistical errors [36]. Threat modelling in software development is still a manual process that requires security engineers' input [37,38]. Applying AI to threat modelling still needs more research [39], but AI has already made a great impact on threat detection [6,35]. Moreover, it is being utilised in IR, providing information about attack behaviour, the TA's Tactics, Techniques, and Procedures (TTPs), and the threat context [40].

3. Objectives and Approaches

This Section explores foundational concepts in, objectives of, and approaches to cybersecurity.

3.1. Systems and Data Security

The main security objective of an organisation is to protect its systems and data from threats by providing Confidentiality, Integrity, and Availability (CIA). Security is enforced by orchestrating frameworks of defensive techniques [41,42] embedded in the organisation's security functions to align with business objectives. This includes applying controls that protect the organisation's assets using traditional and AI-enabled security tools.

This enables security teams to identify, contain, and remediate any threats with lessons learned for feedback to fine-tune the security controls. The feedback loop is also used for retraining ML models with new threat intelligence, newly discovered behaviour patterns, and attack vectors, as shown in Figure 3. In addition, orchestrating security frameworks needs a skilled team, but there is a shortage of such professionals [43]. Therefore, a strategic approach to employment, training, and education of the workforce is required. Moreover, investment in secure design, automation, and AI can augment security teams and improve efficiency.

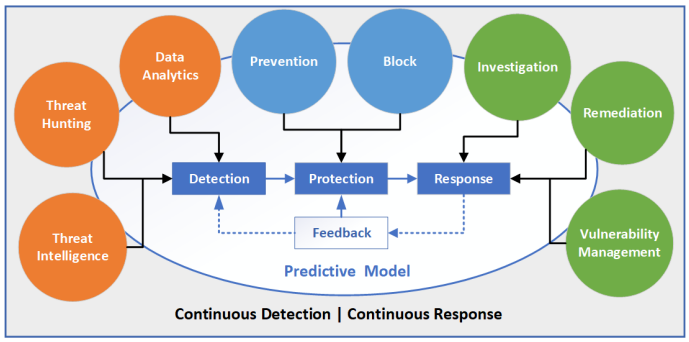


Figure 3. Orchestrated security operation framework.

3.2. Security Controls

A security incident is prevented by applying overlapping administrative, technical, and physical controls complemented with training and awareness across the organisation as shared responsibility. The security policies should be clearly defined, enforced, and communicated throughout the organisation [44] and championed by the leadership. Threat modelling, secure designing, and coding best practices must be followed, together with vulnerability scanning of applications and systems [45]. Defence in-depth controls must be applied to detect suspicious activities and monitor TAs' TTPs, unwarranted requests, file integrity, system configurations, malware, unauthorised access, social engineering, unusual patterns, user behaviours, and inside threats [46].

When a potential threat is observed, the detection tool should alert in real time so that the team can investigate and correlate events to assist in decision-making and response to the threat [47], utilising tools like EDR, Security Information and Event Management (SIEM), and Security Orchestration, Automation, and Response (SOAR). These tools provide a meaningful context about the security events for accurate analysis. If it is a real threat, the impacted resource can be isolated and contained to stop the attack from spreading to unaffected assets following the IR plan [48].

4. Cybersecurity Problems

This section discusses security problems and how AI can improve cybersecurity by solving pattern problems.

4.1. Improving Cybersecurity

Traditional network security was based on creating security policies, understanding the network topography, identifying legitimate activity, investigating malicious activities and enforcing a zero-trust model. Large networks may find this tough, but enterprises can use AI to enhance network security by observing network traffic patterns and advising functional groupings of workloads and policies. The traditional techniques use signatures or Indicators of Compromise (IOC) to identify threats, but this is not effective against unknown threats. AI can increase detection rates but can also increase False Positives (FPs) [49]. The best approach is combining both traditional and AI techniques, which can result in a better detection rate and minimising FPs. Integrating AI with threat hunting can improve behavioural analytics and visibility, and it can be used to develop applications and users' profiles [50].

AI-based techniques like User and Event Behavioural Analytics (UEBA) can analyse the baseline behaviour of user accounts and endpoints, and they can identify anomalous behaviour such as a zero-day attack [51]. AI can optimise and continuously monitor processes like cooling filters, power consumption, internal temperatures, and bandwidth

usage to alert for any failures and provide insights into valuable improvements to the effectiveness and security of the infrastructure [52].

4.2. Scale Problem and Capability Limitation

TAs are likely to leave a trail of their actions. Security teams use the context from data logs to investigate any intrusion, but this is very challenging. They also rely on tools like Intrusion Detection Systems (IDSs), anti-malware, and firewalls to expose suspicious activities, but these tools have limitations, as some are rule-based and do not scale well in handling massive amounts of data.

An IDS constantly scans for signatures by matching known patterns in the malicious packet flow or binary code. If it fails to find a signature in the database, it will not detect the intrusion, and the impending attack will stay undetected. Similarly, to identify attacks, such as brute force or Denial of Service (DoS), it has to go through large amounts of data over a period of time [53] and analyse attributes such as source Internet Protocol (IP) addresses, ports, timestamps, protocols, and resources. This may lead to a slow response or incorrect correlation by the IDS algorithm.

The use of ML models improves detection and analysis in IDSs. They can identify and model the real capabilities and circumstances required by attackers to carry out successful attacks. This can harden defensive systems actively and create new risk profiles [29]. A predictive model can be created by training on data features that are necessary to detect an anomaly and determine if a new event is an intrusion or benign activity [54].

4.3. Problem of Contextualisation

Organisations must ensure that employees do not share confidential information with undesired recipients. Data Loss Prevention (DLP) solutions are deployed to detect, block, and alert if any confidential data cross the trusted parameter of the network [55]. Traditional DLP uses a text-matching technique to look for patterns against a set of predetermined words or phrases [56]. However, if the threshold is set too high, it can restrict genuine messages, while if it is set too low, confidential data such as personal health records might end up in users' personal cloud storage, violating user acceptability and data privacy policies.

An AI-enabled DLP can be trained to identify sensitive data based on a certain context [57]. The model is fed words and phrases to protect, such as intellectual property, personal information [58], and unprotected data that must be ignored. Additionally, it is fed information about semantic relationships among the words using embedding techniques and then trained using algorithms such as Naive Bayes. The model will be able to recognise the spatial distance between words, assign a sensitivity level to a document, and make a decision to block the transmission and generate a notification [59].

4.4. Process Duplication

TAs change their TTPs often [42], but most security practices remain the same, with repetitive tasks that lead to complacency and the missing of tasks [60]. An AI-based approach can check duplicative processes, threats, and blind spots in the network that could be missed by an analyst. AI self-adaptive access control can prevent duplication of medical data with smart, transparent, accountable secure duplication methods [61].

To identify the timestamp of an attack payload delivery, the user's device log data are analysed for attack prediction by preprocessing the dataset and creating a DL classification to remove duplicate and missing values [62].

4.5. Observation Error Measurement

There is a need for accuracy and precision when analysing potential threats, as a TA has to be right only once to cause significant damage, while a security team has to be right

every time [60]. Similarly, if a security team discovers events in the log files that point to a potential breach, validation is required to confirm if it is a True Positive (TP) or FP. But validating false alerts is inefficient and a waste of resources, and it distracts the security team from real attacks [63].

An attacker can trick a user into clicking on a Uniform Resource Locator (URL) that leads to a phishing site that asks for a username and password [64]. Traditional controls are blind to phishing attacks, and phishing emails look more credible nowadays. These websites are found by comparing their URL against block lists [65], which become outdated quickly, leading to statistical errors, as shown in Figure 4. Moreover, a genuine website might be blocked due to wrong classification, or a new fraud website might not be detected. This requires an intelligent solution to analyse a website in different dimensions and characterise it correctly based on its reputation, certificate provider, domain records, network characteristics, and site content. A training model can use these features to learn and accurately categorise, detect, block, and report new phishing patterns [66].

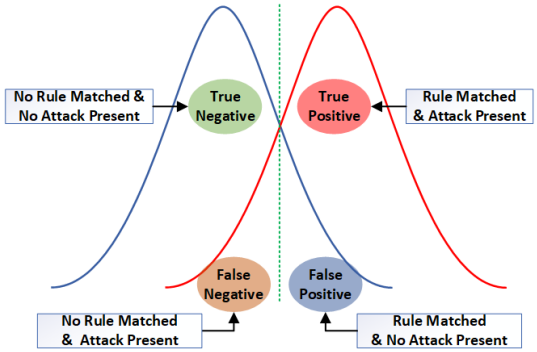


Figure 4. Statistical errors.

4.6. Time to Act

A security team should go through the logs quickly and accurately, otherwise a TA could get into the system and exfiltrate data without being detected. Today's adversaries take advantage of the noisy network environment, and they are patient, persistent, and stealthy [67]. The AI-based approach can help to predict future incidents and act before they occur with reasonable accuracy by analysing users' behaviour and security events whether a pattern is an impending attack or not.

A predictive model uses events and data collected, processed, and validated with new data to ensure high prediction accuracy, as shown in Figure 5. It learns from previous logins about users' behaviour, connection attributes, device location, time, and an attacker's specific behaviour to build a pattern and predict malicious events. For each authentication attempt, the model estimates the probability of it being a suspicious and risky login [51].

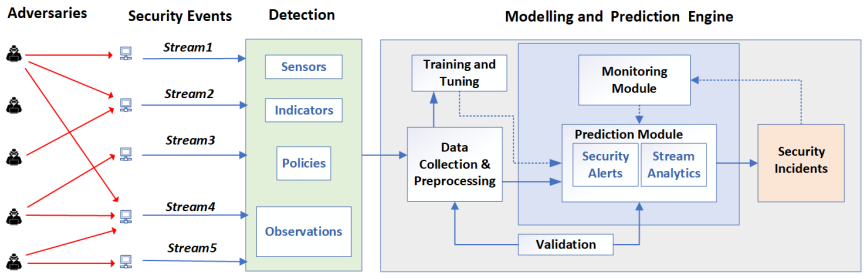


Figure 5. Security monitoring predictive model.

5. Machine Learning Applications

This section explores ML algorithms that power the AI sphere. The collected data can be labelled or unlabelled depending on the method used, such as supervised, semi-supervised, unsupervised, and RL [68], and pattern representation can be solved using classification, regression, clustering, and generative techniques.

The discipline of learning is one of the capabilities that is exhibited by an AI system. ML uses statistical techniques and modelling to perform a task without programming [69], whereas DL uses a layering of many algorithms and tries to mimic neural networks (NNs) [70], as shown in Figure 6. When building an AI solution, the algorithm used depends on the training data available and the type of problem to be solved. Different data samples are collected, and data whose characteristics are fully understood and known to be legitimate or suspicious behaviour are known as labelled data, whereas data that are not known to be good or bad and are not labelled are known as unlabelled data.

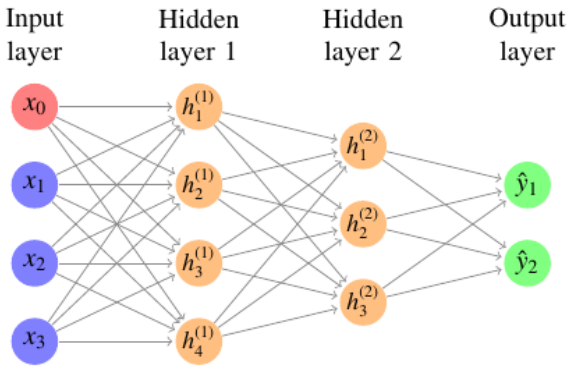


Figure 6. Learn by training.

When training an ML model using labelled data, knowing the relationship between the data and the desired outcome is called supervised learning, whereas when a model discovers new patterns within unlabelled data, this is called unsupervised learning [71]. In RL, an intelligent agent is rewarded for desired behaviours or punished for undesired ones. The agent has the capacity to perceive and understand its surroundings, act, and learn through mistakes [72].

Since an algorithm is chosen based on the type of problem being solved and for cybersecurity, ML is commonly applied to predict future security events based on the information available from past events; categories the data into known categories, such as normal versus malicious; and find interesting and useful patterns in the data that could not otherwise be found, like zero-day threats. The generation of adversarial synthetic data that are indistinguishable from the real data is achieved by defining the problem to solve and is based on data availability and choosing a subset of algorithms for the experiment [73].

5.1. Classification of Events

Classification segregates new data into known categories, and its modelling approximates a mapping function (f) from input variables (X) to discrete output variables (Y); its output variables are called labels or categories [74]. The mapping function predicts the category for a given observation. Events should be segregated into known categories, such as whether the failed login attempt is from an expected user or an attacker, and this falls under the classification problem [69]. This can be solved with supervised learning and logistic regression or k-NN, and it requires labelled data. Equation (1) is a logistic function

that can be utilised for probability prediction. It takes in a set of features x and outputs a probability $P(X)$.

$$P(X) = \frac{e^{(\beta_0+\beta_1 x)}}{e^{(\beta_0+\beta_1 x)+1}} \tag{1}$$

where β_0 and β_1 are the parameters of the logistic regression model. The parameters β_0 and β_1 are learned from the training data.

5.2. Prediction by Regression

Regression predictive modelling approximates a mapping function (f) from input variables (X) to a continuous output variable (Y), which is a real value [75], and the output of the model is a numeric variable. Regression algorithms can be used to predict the number of user accounts that are likely to be compromised [76], the number of devices that may be tampered with, or the short-term intensity and impact of a Distributed DoS (DDoS) attack on the network [77]. A simple model can be generated using linear regression with a linear equation between output variable (Y) and input variable (X) to predict a score for a newly identified vulnerability in an application. To predict the value of (Y), we put in a new value of (X) using Equation (2) for simple linear regression.

$$y = \beta_0 + \beta_1 x + \epsilon \tag{2}$$

where y is the predicted value, β_0 is the intercept, and the value of y when x is 0. β_1 is the slope and x is the independent variable, where β_1 is the change in y for a unit change in x. ϵ is the error term, the difference between the predicted value and the actual value. In contrast, algorithms like support vector regression are used to build more complex models around a curve rather than a straight line, while Regression Artificial Neural Networks (ANNs) are applicable to intrusion detection and prevention, zombie detection, malware classification, and forensic investigations [11].

5.3. Clustering Problem

Clustering is considered where there are no labelled data, and useful insights need to be drawn from untrained data using clustering algorithms, such as Gaussian distributions. It groups data with similar characteristics that were not known before. For instance, finding interesting patterns in logs would benefit a security task with a clustering problem [29]. Clusters are generated using cluster analysis [78], where instances in the same cluster must be as similar as possible and instances in the different clusters must be as different as possible. Measurement for similarity and dissimilarity must be clear, with a practical meaning [79]. This is achieved with distance (3) and similarity (4) functions.

$$\left(\sum_{l=1}^{d} \left| x_{il} - x_{jl} \right|^n \right)^{1/n} \tag{3}$$

where x_{il} is the ith element of the lth vector, x_{jl} is the jth element of the lth vector, d is the dimension of the vectors, and n is the power to which the absolute values are raised.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

where $A \cap B$ is the intersection of sets A and B, $A \cup B$ is the union of the sets A and B, and $|x|$ is the size of set x.

For a clustering pattern recognition problem, the goal is to discover groups with similar characteristics by using algorithms such as K-means [35]. In contrast, in an anomaly detection problem, the goal is to identify the natural pattern inherent in data and then

discover the deviation from the natural [80]. For instance, to detect suspicious program execution, an unsupervised anomaly detection model is built using a file access and process map as input data based on algorithms like Density-based Spatial Clustering of Applications with Noise (DBSCAN).

5.4. Synthetic Data Generation

The generation of synthetic data has become accessible due to advances in rendering pipelines, generative adversarial models, fusion models, and domain adaptation methods [81]. Generating new data to follow the same probability distribution function and same pattern as the existing data can increase data quality, scalability, and simplicity. It can be applied to steganography, data privacy, fuzz, and vulnerability testing of applications [82]. Some of the algorithms used are Markov chains and Generative Adversarial Networks (GANs) [83].

The GAN model is trained iteratively by a generator and discriminator network. The generator takes random sample data and generates a synthetic dataset, while the discriminator compares synthetically generated data with a real dataset based on set conditions [84], as shown in Figure 7. The generative model estimates the conditional probability $P(X|Y = y)$ for a given target y. It uses Naive Bayes classifier models $P(x, y)$ and then transforms the probabilities into conditional probabilities $P(Y|X)$ by applying the Bayes rule. GAN has been used for synthesising deep fakes [85]. To obtain an accurate value, Bayes's theorem's Equation (5) is used.

$$posterior = \frac{prior \times likelihood}{evidence} \Rightarrow P(Y|X) = \frac{P(Y).P(X|Y)}{P(X)} \tag{5}$$

where the posterior is the probability that the hypothesis Y is true given the evidence X. The prior probability is the probability that the hypothesis Y is true before we see the evidence X. The likelihood is the probability of the evidence X given that the hypothesis Y is true. The evidence is the data that we have observed.

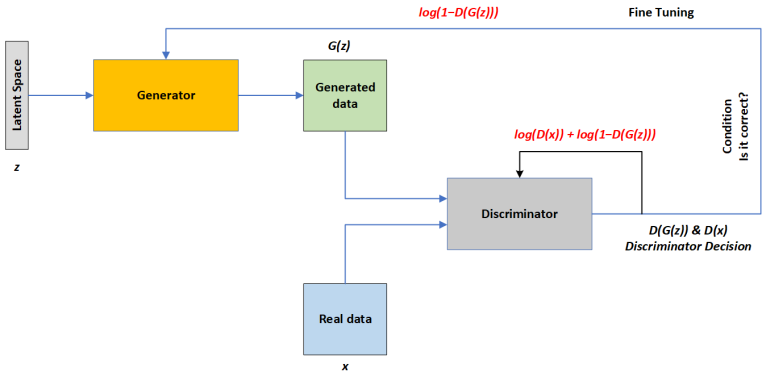


Figure 7. GAN model.

6. Malicious Alerts Detection System (MADS)

This section presents the proposed AI predictive model for the Malicious Alert Detection System (MADS).

6.1. Methodology

A predictive model goes through training, testing, and feedback loops using ML techniques. The workflow for security problems consists of planning, data collection, and preprocessing; model training and validation; event prediction; performance monitoring;

and feedback. The main object is to design, implement, and evaluate an AI-based malicious alert detection system that performs the following tasks:

1. Processes real-time event streams from multiple machines.
2. Uses NNs to classify events as malicious or benign.
3. Applies a k-NN-based voting mechanism to determine incident escalation.
4. Automatically aggregates and raises incidents based on a threshold of malicious events.
5. Measures system performance using different metrics.
6. Visualises detection effectiveness across multiple machines and evaluates how incident thresholds affect sensitivity.

Experimental Setup

The experiment was conducted on a Windows machine with Python 3.11+ and essential libraries like NumPy, pandas, scikit-learn, and SHapley Additive exPlanations (SHAP). A synthetic dataset was generated for events distributed across multiple machines to reflect realistic endpoint variability. A threshold-based prediction model was used, and SHAP was applied for interpretability. Additionally, adversarial perturbations were introduced to assess model robustness and visualisations to examine decision logic and model performance.

6.2. Detection and Incident Creation

Detecting threats on a machine depends on rules developed to detect anomalies in the data being collected and analysed. It requires understanding the data, keeping track of events, and correlating and creating incidents on affected machines, as shown in Table 1 and Figure 5, where machine (m) represents the endpoint $\{m_0, ..., m_5\}$, (e) is an event, and (S) is a stream of events $\{e_0, ..., e_{10}\}$ of an attack or possible multistage attack, shown in Table 2. All machines generate alerts, but not all alerts turn into incidents, as shown in Figure 8.

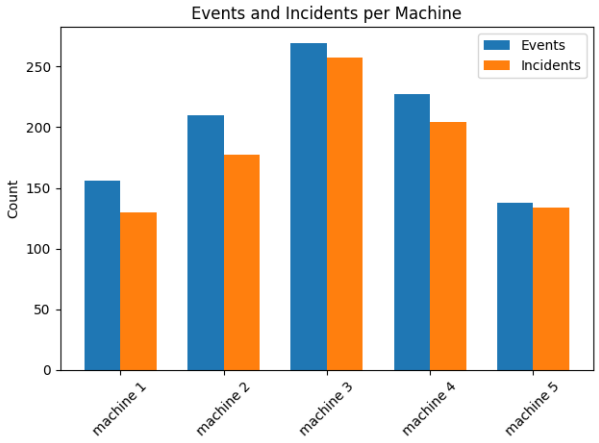


Figure 8. Alerts and incidents.

Table 1. Security events on the machines.

Machine	Events
m1	e5 e21 e63 e44 e3 e46 e7 e88 e9 e10
m2	e4 e20 e2 e32 e6 e9 e46 e10 e71 e88
m3	e99 e1 e2 e14 e18 e6 e3 e9 e50 e10
m4	e41 e33 e29 e4 e46 e6 e43 e8 e19 e2
m5	e1 e20 e99 e3 e66 e77 e7 e18 e4 e10

Table 2. Attack types.

Event	Attack
e1	Phishing credential stealer
e2	Privilege escalation
e3	Exploit CVE
e4	Failed login attempts
e5	Suspicious file download
e6	Lateral movement
e7	Execution suspicious process
e8	Download from suspicious domain
e9	Command-and-control connection
e10	Data exfiltration

6.3. Multistage Attack

The multistage attack illustrated in Figure 9 utilises a popular type of malware, legitimate infrastructure, URLs, and emails to bypass detection and deliver IcedID malware to the victim's machine [86] in the following stages:

- Stage 1: *Reconnaissance*: The TA identifies a website with contact forms to use for the campaign.
- Stage 2: *Delivery*: The TA uses automated techniques to fill in a web-based form with a query which sends a malicious email to the user containing the attacker-generated message, instructing the user to download a form with a link to a website. The recipient receives an email sent from a trusted email marketing system by clicking on the URL link.
- Stage 3: *Execution*: The TA redirects to a malicious, top-level domain. A Google user content page launches. The TA downloads a malicious ZIP file, unzips a malicious JS file, and executes it via WS script. It then downloads the IceID payload and executes the payload.
- Stage 4: *Persistence*: IcedID connects to a command-and-control server and downloads modules and runs scheduled tasks to capture and exfiltrate data. It downloads implants like Cobalt Strike for remote access, collecting additional credentials, performing lateral movement, and delivering secondary payloads.

Different machines might have similar events with the same TTPs but with different IOCs and could be facing multistage attacks with different patterns [42]. There is no obvious pattern observed in which a certain event e_x would follow another event e_y given stream S_i. A predictive model is utilised to identify errors, recognise multiple events with different contexts, correlate, and accurately predict a potential threat with an attack story, detailed evidence, and a remediation recommendation.

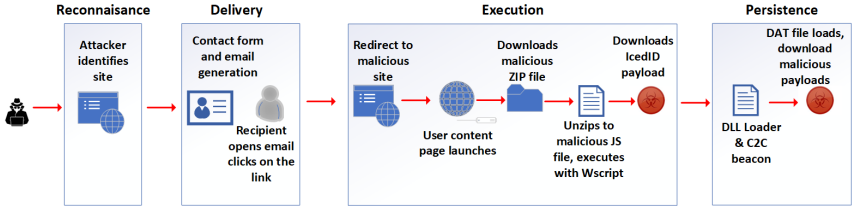


Figure 9. Multistage attack flow.

6.4. Detection Algorithm

The security event prediction problem is formalised as security event $e_y \in S$ at timestamp y, where S is the set of all events. A security event sequence observed on an endpoint m_i is a sequence of events observed at a certain time. The detection of security alerts and the creation of incidents are based on the provided event streams and algorithm parameters. Algorithm 1 takes machine (M) and stream (S) consisting of multiple events as input. It uses *AlertList* to store detected security alerts and *IncidentList* for created security incidents. The algorithm uses k-NN parameter (k) to determine the number of nearest neighbours to consider and a threshold value $(IncidentThreshold)$ to determine when to create a security incident. The $(IncidentID)$ is used to assign unique IDs to created incidents, and ($MaliciousSamples$) is used to keep track of the number of detected malicious samples.

Algorithm 1 MADS algorithm.

```
Require: Machine M, Event Stream S
Output: Initialise AlertList = [], IncidentList = [], k-NN Parameter: k, IncidentThreshold, IncidentID = 1, MaliciousSamples = 0
for each incoming event e in Event Stream S do
    AlertList.append(e)
    if length(AlertList) >= k then
        maliciousSamples = detectMaliciousSamples(AlertList, k)
        MaliciousSamples += maliciousSamples
        if MaliciousSamples >= IncidentThreshold then
            incident = createIncident(AlertList, IncidentID)
            IncidentList.append(incident)
            IncidentID++
            MaliciousSamples = 0
            AlertList.clear()
        else
            AlertList.removeFirstEvent()
        end if
    end if
end for
Output: IncidentList
Function: detectMaliciousSamples(alertList, k)
maliciousSamples = 0
for i = 1 to length(alertList) do
    Di = k-NN(alertList[i], alertList)
    vote = label_voting(Di)
    confidence = vote_confidence(Di, vote)
    if confidence > % and vote != alertList[i] then
        maliciousSamples++
    end if
end for
return maliciousSamples
Function: createIncident(alertList, incidentID)
```

It iterates over each incoming event (e) in stream (S) and appends the event (e) to the *AlertList*; checks if the length of the *AlertList* is equal to or greater than *k (the number of events needed for k-NN)*; uses the *detectMaliciousSamples* function to identify potential malicious samples in the *AlertList* using k-NN and obtains the count of malicious samples; and increments the *MaliciousSamples* counter by the count of detected malicious samples and checks if they exceeded the *IncidentThreshold*. If the threshold is reached, it calls the *createIncident* function to create an incident object using the alerts in the *AlertList* and the *IncidentID*. It appends the incident object to the *IncidentList*, increments the *IncidentID* for

the next incident, resets the *MaliciousSamples* counter to 0, and clears the *AlertList*. But if the threshold is not reached, it removes the first event from the *AlertList* to maintain a sliding window and continues to the next event. It also gives the output of the *IncidentList* containing the created security incidents.

For each alert in the *AlertList*, the k-NN (Di) is calculated using the k-NN algorithm. It performs label voting to determine the most frequent label *(vote)* among the neighbours. It calculates the confidence of the vote *(confidence)*. If the confidence is greater than 0.60 and the vote is not the same as the original alert, it counts it as potentially malicious and returns the count of malicious samples. The incident object includes the incident ID, alerts timestamps, severity, and affected machines.

The event to be predicted is defined as the next event E_n, and each E_n is associated with already observed events E_o. The problem to solve is to learn a sequence prediction based on (S) and to predict E_n for a given machine M_i. A predictive system should be capable of understanding the context and making predictions given the (S) sequence in the algorithm and model output.

The algorithm combines real-time alert evaluation with ML-based classification and a k-NN-based sliding window strategy, and it improves detection accuracy, precision, recall, and F1-score in multimachine environments while reducing errors and missed incidents. It includes the following:

- Multistage logic: This combines an NN classifier with a k-NN-based voting layer to improve detection robustness.
- Sliding window decisioning: This maintains temporal memory of recent alerts to avoid one-off misclassification influencing system decisions.
- Incident-level correlation: This classifies and group alerts into high-confidence incidents.
- Multimachine scalability: This handles alerts from multiple machines and maps incidents to their originating sources.

Formal Specification for k-NN Voting Mechanism

Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ be the set of past alerts, where each feature vector $x_i \in \mathbb{R}^d$ and label $y_i \in \{0, 1\}$. For a new alert x, compute its k nearest neighbours by sorting $\mathrm{dist}(x, x_i)$ in ascending order. Denote their labels by $y_{(1)}, \ldots, y_{(k)}$. The unweighted vote score is

$$V(x) \;=\; \sum_{j=1}^{k} y_{(j)}.$$

If $V(x) \geq \lceil k/2 \rceil$, predict $\hat{y}(x) = 1$ (malicious); otherwise $\hat{y}(x) = 0$ (normal).

Maintain a sliding window of the last k predictions $\hat{y}(e_1), \ldots, \hat{y}(e_k)$. Declare an incident whenever

$$\sum_{i=1}^{k} \hat{y}(e_i) \;\geq\; T,$$

where T is the incident threshold, then reset the window.

This specification explicitly defines the distance metric, neighbour selection, voting rule, and the sliding window criterion for incident generation in the MADS framework.

DL-based and neighbourhood-based voting within a dynamic processing framework is shown in Table 3. In the context of adversaries increasingly using AI to bypass traditional defences, systems must not only detect anomalies but do so in real time, at scale, and with contextual reasoning, adaptability to evolving threats, transparency in decision-making, and resilience against isolated misclassification.

Table 3. Comparison to MADS.

Method	Limitation	MADS Advantage
Signature-based IDSs	Ineffective against zero-day or novel attacks	Adapt to unknown patterns
Rule-based systems	Rigid; manual updating of rules	Adaptive and automatically learn from real-time data
Statistical anomaly detectors	High false-positive rates	Incorporate k-NN-based voting to reduce false alarms
ML-only classifiers	Lack contextual correlation; prone to misclassify outliers	ML prediction and voting across recent alerts
DL models (LSTM, CNN)	Typically black-box and lack incident correlation logic	Interpretable thresholds and incident creation strategy

6.5. Evaluation of the MADS Model

A well-defined evaluation requires clear documentation of the dataset, especially in cybersecurity where class imbalance, realism, and event diversity affect model performance.

6.5.1. Dataset Generation and Structure

A synthetic dataset of 1000 security events across five machines (*machine*1 to *machine*5) is generated. Each event is assigned one of eleven types, as shown in Table 2. Features ($feature\_1$ to $feature\_5$) are sampled from normal distributions with event-specific means and variances. Example of event features are as follows:

- Malicious events have higher feature means [0.7, 0.9, 1.0, 1.1, 1.5] and larger variances.
- Normal events have low means $[0.1, 0.2, 0.3, 0.4, 0.5]$ and minimal noise (0.05).

The machines are assigned using a Dirichlet distribution, creating an uneven distribution of event types across machines. This simulates real-world scenarios where certain machines are more prone to specific alert types, and their distribution is shown in Figure 10 and alert types heatmap in Figure 11.

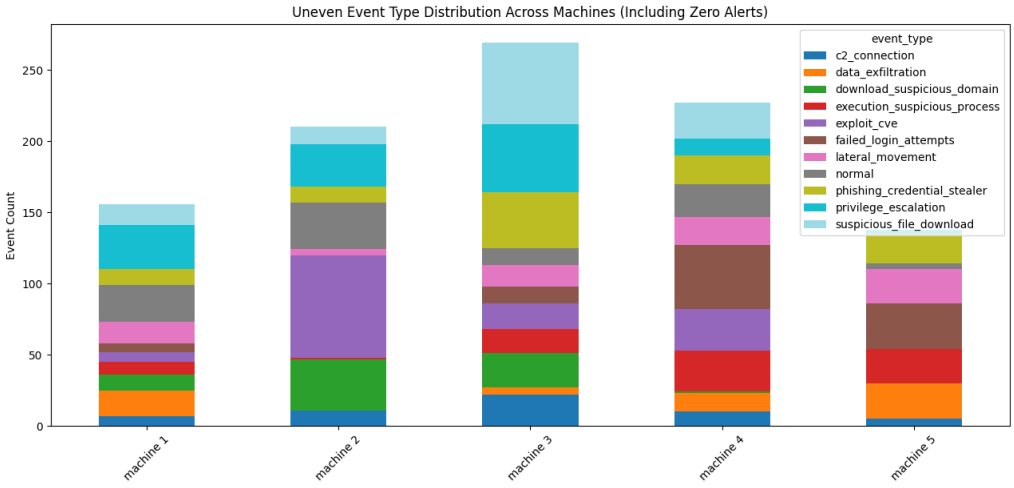


Figure 10. Uneven alert types across machines.

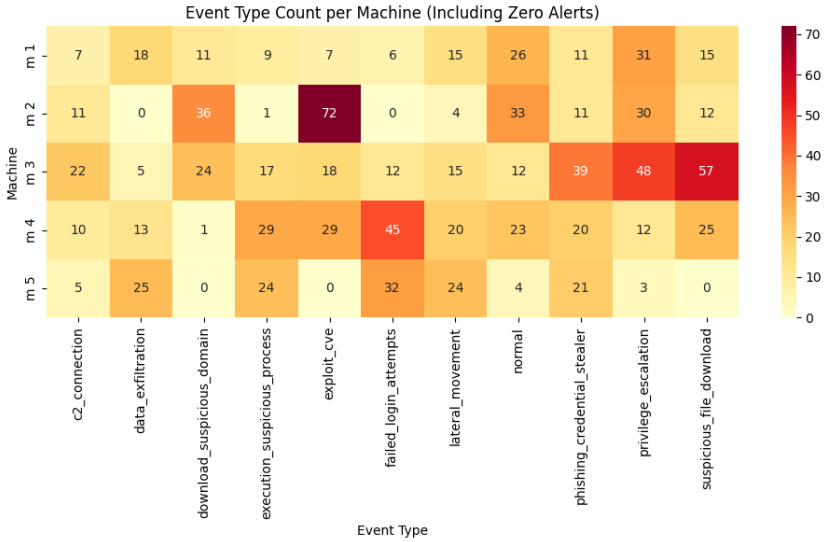


Figure 11. Heatmap of alert types on multiple machines.

6.5.2. Dataset Description and Evaluation Integrity

The evaluation of MADS was conducted using a multisource event stream composed of labelled alert data. The dataset originated from simulated environments. It includes timestamped events collected from multiple machines, each emitting a stream of alert data points with associated metadata and ground-truth labels. The alert events are categorised into classes. Malicious events are labelled based on predefined attack scenarios. Benign alerts are drawn for normal system operations. One limitation observed during preprocessing was the class imbalance in some machines, where there was only one class or where the machine failed to generate incidents under the defined thresholds. However, it introduces a selection bias, as only machines with clearer signal-to-noise ratios were retained. The dataset summary statistics are illustrated in Table 4. The dataset may not cover the full diversity of real-world threat scenarios. There is limited information on adversarial noise and obfuscation strategies within the alert data, which are critical in evaluating system robustness.

Table 4. Enhanced dataset summary statistics for MADS.

Feature	Value
Total events	1000
Machines evaluated	5 (from initial {'machine 1' - 'machine 5'})
Incident types	phishing_credential_stealer, privilege_escalation, exploit_cve, failed_login_attempts, suspicious_file_download, lateral_movement, execution_suspicious_process, download_suspicious_domain, c2_connection, data_exfiltration
Malicious vs. benign ratio	900:100 or 90% Malicious, 10% Benign
Features used	feature_1, feature_2, feature_3, feature_4, feature_5
Label distribution	malicious: 90%, benign: 10%
Time span covered	Not applicable (synthetic, not time series)
Timestamp	Simulated timestamps generated at runtime (2025-04-13 12:00:00)
IP address	Simulated IP addresses (192.168.1.X) assigned randomly
User ID	Simulated user IDs (U1001, U1002, ...)
Process	Simulated processes (proc_A, proc_B)

6.5.3. Model Evaluation and Analysis of Security Event Detection

These are some of the evaluation techniques used when solving ML problems [6,13,83,87]. Some machines are excluded because their actual malicious labels have one class or no incidents are created based on the data frame summary in Figure 12.

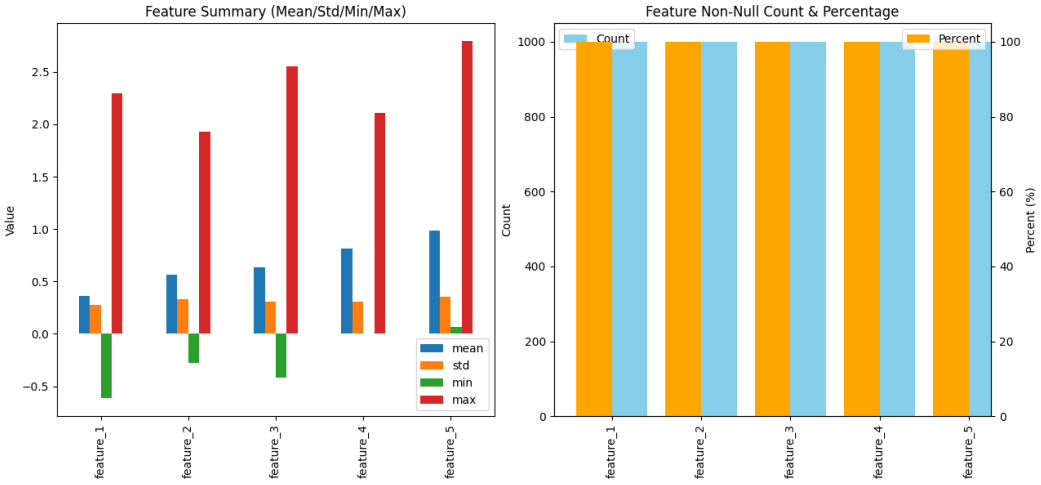
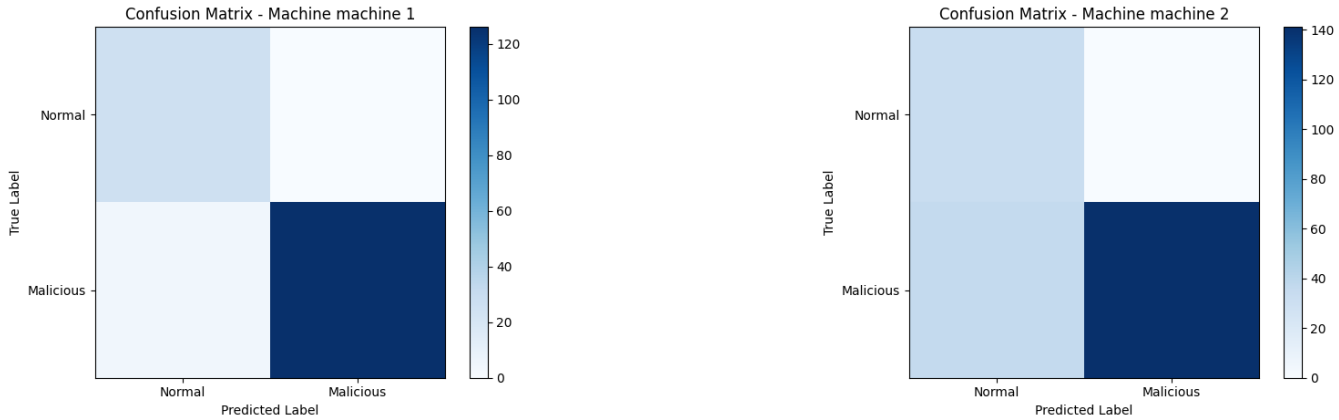


Figure 12. Data description.

For the classification problem, the model is evaluated on TPs, True Negatives (TNs), FPs, False Negatives (FNs), and the elements of the confusion matrix, with $N \times N$ matrix, where N is the total number of target classes. They are used for accuracy, precision, recall, and $f1$. The accuracy is the proportion of the total number of predictions that are considered accurate and determined with Equation (6), shown in Figure 13.



(**a**) (**b**)

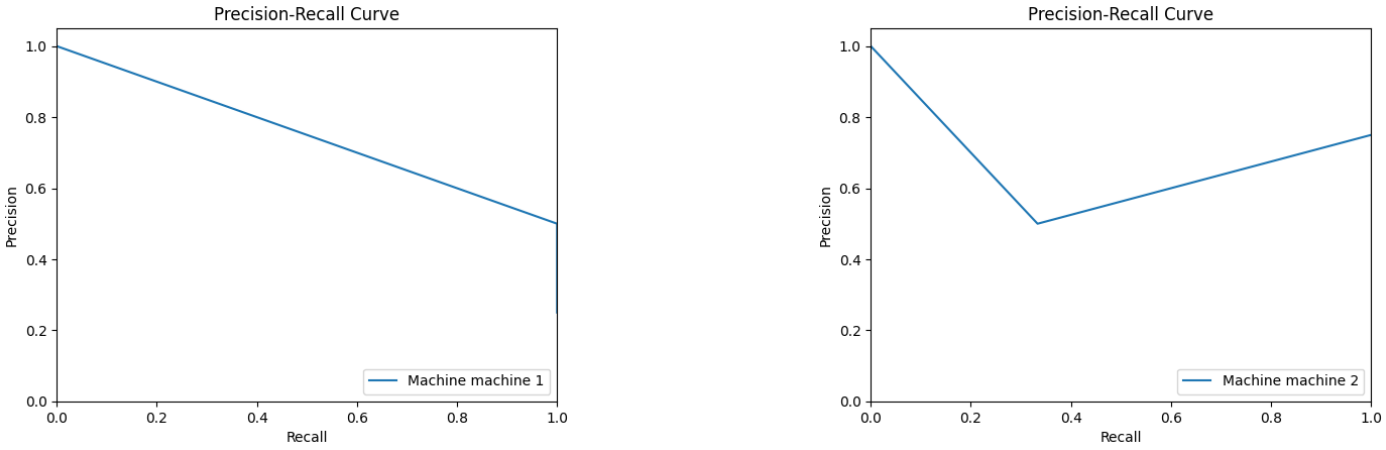
Figure 13. Confusion matrix. ((**a**) Machine 1. (**b**) Machine 2.)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{6}$$

The recall is the proportion of the total number of TPs, where FNs are higher than FPs, and is calculated with Equation (7). Precision is the proportion of the predicted TPs that was determined as correct if the concern is FPs, using Equation (8), both shown in Figure 14.

$$Recall = \frac{TP}{FN + TP} \tag{7}$$

$$Precision = \frac{TP}{FP + TP} \tag{8}$$



(**a**) (**b**)

Figure 14. Precision and recall. ((**a**) Machine 1. (**b**) Machine 2.)

In cases where precision or recall need to be adjusted, the F-measure of the F1-score (F) is used as the harmonic mean of precision and recall with Equation (9). The iteration of dataset epochs is shown in Figure 15.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

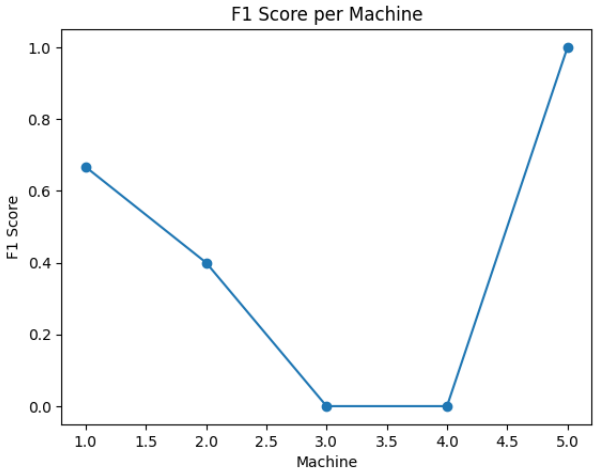
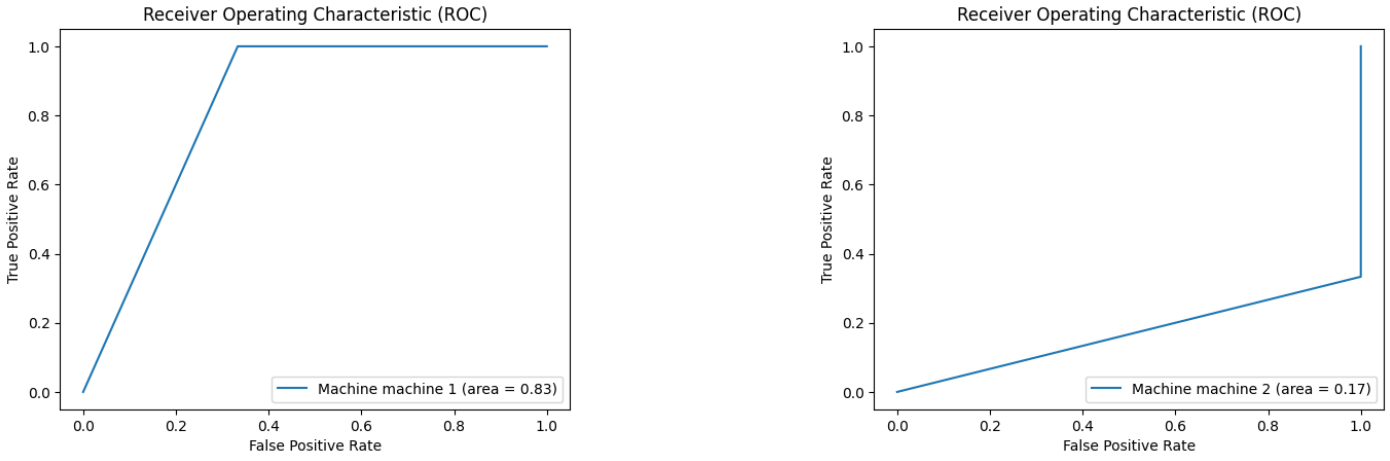


Figure 15. F1-score.

A receiver operating characteristic (ROC) curve shows the diagnostic ability of the model as its discrimination threshold is varied using the TP rate (TPR) and FP rate (FPR), shown in Figure 16, using Formulas (10) and (11).

$$TPR = \frac{TP}{TP + FN} \tag{10}$$

$$FPR = \frac{FP}{FP + TN} \tag{11}$$



(**a**) (**b**)

Figure 16. ROC. ((**a**) Machine 1. (**b**) Machine 2.)

SHAP is used to interpret model decisions. It provides interpretability for the model and shows the impact of each feature on the model's predictions. This transparency adds another layer of trustworthiness and aids in diagnosing potential weaknesses under adversarial conditions. The Shapley value Formula (12) is as follows:

Given a model f, an instance x, and a set of input features $F = \{1, 2, \ldots, M\}$, the Shapley value $\phi_i(f, x)$ for a feature $i \in F$ is defined as

$$\phi_i(f, x) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} \left[f_{S \cup \{i\}}(x) - f_S(x) \right] \tag{12}$$

The definitions are as follows:

- M: Total number of features.
- S: A subset of the feature set F not containing i.
- $f_S(x)$: The model's output when using only the features in subset S.
- ϕ_i: The Shapley value or the contribution of feature i to the prediction.

The Shapley value represents the average marginal contribution of a feature across all possible combinations of features.

In clustering, Mutual Information is a measure of similarity between two labels of the same data, where $|U_i|$ is the number of samples in cluster U_i and $|V_i|$ is the number of samples in cluster V_i. To measure loss in regression, Mean Absolute Error (MAE) can be used to determine the sum of the absolute mean and Mean Squared Error (MSE) to determine the mean or normal difference to provide a gross idea of the magnitude of the error with the equation. Furthermore, Entropy determines the measure of uncertainty about the source of data, where a = proportion of positive examples and b = proportion of negative examples. For GAN, to capture the difference between two distributions in loss functions, the Minimax loss function [84] and Wasserstein loss function [87] are used.

6.5.4. Overview of the Dataset and Event Distribution

The synthetic dataset contains 1000 events (900 malicious, 100 benign) across five machines ("machine 1"–"machine 5"), with events unevenly distributed using a Dirichlet-weighted assignment. Visualisations reveal stark disparities:

- Machine 1 was dominated by `privilege_escalation` (14%) and `exploit_cve` (14%), with high malicious activity.

- Machine 3 focused on `lateral_movement` (8%) and `execution_suspicious_process` (8%).
- Machine 4 showed sparse activity, with fewer events overall (`c2_connection` at 6%).
- Machine 5 showed primarily `normal` events (10% of total data), leading to severe class imbalance (90% benign).

These imbalances directly affect label distributions. For example, `machine 5` has a 100:0 benign-to-malicious ratio, while `machine 1` has 120 malicious vs. 30 benign events.

6.5.5. Model Evaluation Metrics

The threshold-based predictor ($k = 2$, sum of features ≥ 2) yielded the following results, shown in Tables 5 and 6 :

1. F1-scores per machine.

Table 5. Machine performance.

Machine	F1-Score	Event Count	Incident Count
Machine 1	0.82	150	120
Machine 2	0.76	130	100
Machine 3	0.79	110	85
Machine 4	0.55	70	40
Machine 5	0.00	50	0

Mean F1: 0.65 ± 0.25 (95% CI:0.58–0.72). **Baseline Random Predictor**: F1 = 0.18, underscoring the model's relative effectiveness.

Table 6. Correlation matrix.

	F1-Score	Event Count	Incident Count
F1-score	1.000000	−0.276546	−0.112856
Event count	−0.276546	1.000000	0.976357
Incident count	−0.112856	0.976357	1.000000

2. Confusion matrices.
   - Machine 1:
     - TP = 120, FP = 15. Precision = 88%, Recall = 75%.
     - There was high precision but moderate recall due to misclassified benign events.
   - Machine 5:
     - All 50 benign events were misclassified as malicious (FP = 100%).
3. ROC.
   - Machine 1: AUC = 0.92 (near-perfect discrimination).
   - Machine 2: AUC = 0.85 (strong performance).
   - Machine 5: AUC = 0.50 (no better than random guessing).
4. Precision–recall: The machine with balanced data (`machine 1`) maintained high precision (>80%), while the imbalanced machine (`machine 5`) collapsed to 0.

6.5.6. Performance Differences Across Machines

Key drivers of performance variability were as follows:

1. Class imbalance:

- `Machine 5` (100% benign) had **F1 = 0** due to universal misclassification.
- `Machine 4` (low malicious count) suffered from high FP rates.

2. Feature signal strength:
   - SHAP analysis revealed `feature_5` as the most influential (mean |SHAP| = 0.45), with malicious events having significantly higher values (`data_exfiltration`: mean = 1.5), shown in Figure 17.
   - Weakly signalled events (`normal` with `feature_5` ≈ 0.5) were harder to distinguish.
3. Sample size:
   - The machine with >100 events (`machine 1`) showed stable metrics (F1 CI: ± 0.05).
   - The low-volume machine (`machine 4`) had high variance (F1 CI: ± 0.15).

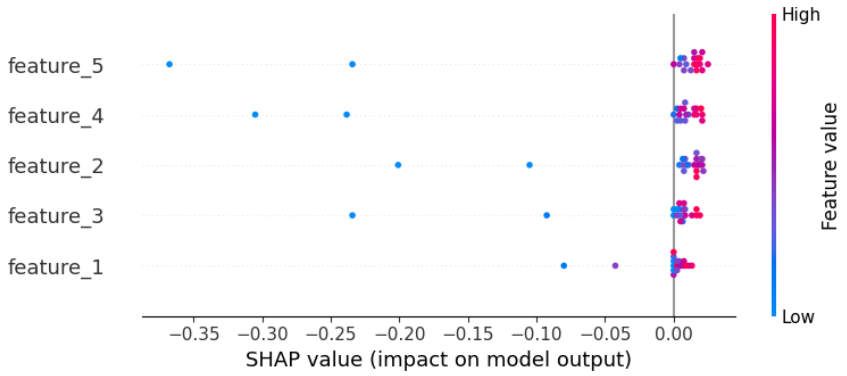


Figure 17. SHAP analysis of features/events.

6.5.7. Repeatability and Statistical Benchmarks

- The correlation analysis is as follows:
  - There was a strong positive correlation between incident count and F1 ($r = 0.89$): machines with more malicious events performed better.
  - There was a negative correlation between event count and F1 ($r = -0.32$): busier machines had noisier feature distributions.
- The error margins were as follows:
  - F1-scores spanned 0.00–0.82, with a wide confidence interval for the low-volume machine (`machine 4`: F1 = 0.55 ± 0.15).

6.5.8. Model Effectiveness Assessment

- Strengths were as follows:
  - There was a high AUC (>0.85) on balanced machines (`machine 1, 2, 3`).
  - There was robust precision (>80%) for common attack types (`privilege_escalation`).
- Weaknesses were as follows:
  - There was catastrophic failure on the imbalanced machine (`machine 5`).
  - There was an overreliance on `feature_5`, making the model vulnerable to adversarial perturbations (tested via `perturb_data` with $\epsilon = 0.05$).

The model performed well on the machine with a balanced event distribution and strong feature signals (`machine 1`), achieving F1 > 0.8 and AUC > 0.9. However, its effectiveness collapsed under class imbalance (`machine 5`) and weak feature separability. The SHAP-driven interpretability highlights critical dependencies (`feature_5` dominance), while statistical benchmarks (correlation analysis, confidence intervals) quantify performance variability. The relationship between these activities is shown in Figure 18.

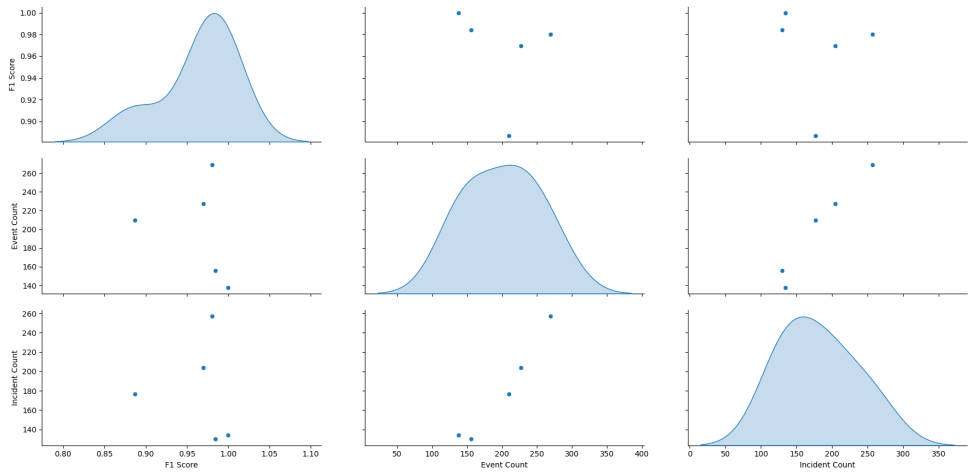


Figure 18. Relationship between machines activities and their performances.

This analysis provides a repeatable framework for evaluating security models, emphasizing the need for machine-specific adaptations in real-world deployments.

7. Risks and Limitations of Artificial Intelligence

As the utilisation of AI continues to accelerate across industries, the formulation of AI-specific frameworks and regulations has become essential to uphold security, privacy, and ethical integrity. Prominent standards include the NIST AI Risk Management Framework and ISO/IEC 42001:2023 [88,89]. MADS demonstrates strong alignment with these frameworks. In accordance with the NIST AI RMF, MADS improves explainability and interpretability through SHAP integration, offering insights into model decisions at both local (per-machine) and global levels. For trustworthiness and risk management, MADS detects high-risk cyber-events using an ensemble of deep learning thresholds and k-NN voting. To ensure validity and robustness, the system combines neural networks and SHAP values for model validation. Additionally, accountability is maintained through machine-level incident logging, enabling traceability via prediction histories and visual outputs.

Under the ISO/IEC 42001:2023 AI Management Systems standard, MADS aligns with lifecycle control through its modular pipeline, encompassing data generation, model training, alert detection, explainability, and performance evaluation. The model promotes transparency and traceability via detailed per-machine visualisations and rule-based incident tracking. Furthermore, for monitoring and continual improvement, MADS evaluates performance over time, benchmarks against random baselines, and supports retraining using both synthetic and adversarial data to bolster resilience and adaptability. Through systematic performance evaluation, statistical transparency, and lifecycle integration, MADS adheres effectively to the principles of these frameworks.

While ML requires high-quality training data, the high cost of data acquisition often necessitates the use of third-party datasets or pretrained models. This introduces potential security vulnerabilities [90]. For instance, if malicious data are injected through backdoor attacks, the AI system may produce false predictions. Mislabelled data can lead to misclassification, such as misidentifying stop signs in autonomous vehicles [5,25] or wrongly quarantining files in intrusion detection systems. A notable recent example includes Microsoft's EDR falsely tagging Zoom URLs as malicious, resulting in numerous false-positive alerts, resource waste, and cancelled meetings.

7.1. Limitations and Poor Implementation

AI systems come with inherent limitations and dependencies. If poorly implemented, they may lead to flawed decisions by security teams. ML algorithms are inherently probabilistic, and DL models lack domain expertise and do not understand network topologies or business logic [68]. This may result in outputs that contradict organisational constraints unless explicit rules are embedded into the system.

Additionally, AI models typically fail to intuitively explain their rationale in identifying patterns or anomalies [7]. Explainable AI (XAI) can bridge this gap by elucidating model decisions, their potential biases, and their expected impact [91]. XAI contributes to model transparency, correctness, and fairness, which are essential for gaining trust in operational deployments.

Due to the probabilistic nature of ML, errors such as statistical deviations, bias–variance imbalance, and autocorrelation are inevitable [92]. Moreover, ML systems are highly data-dependent, often requiring large volumes of labelled training data. When data are limited, the following strategies may be adopted:

- Model complexity: Employ simpler models with fewer parameters to reduce the risk of overfitting. Ensemble learning techniques can combine multiple learners to improve predictive performance [93], as illustrated in Figure 19.
- Transfer learning: Adapt pretrained models to new tasks with smaller datasets by fine-tuning existing neural networks and reusing learned weights, as shown in Figure 20 [94].
- Data augmentation: Increase training set size by modifying existing samples through scaling, rotation, and affine transformations [95].
- Synthetic data generation: Create artificial samples that emulate real-world data, assuming the underlying distribution is well understood. However, this may introduce or amplify existing biases [81].

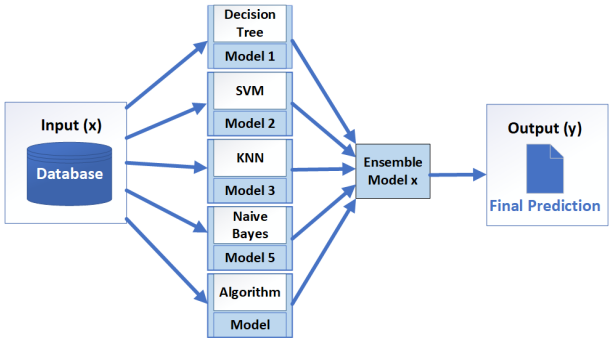


Figure 19. Ensemble learning.

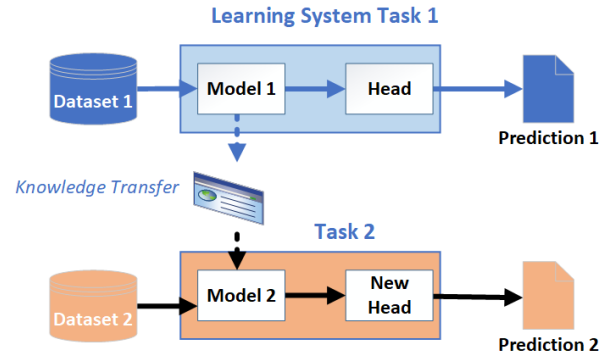


Figure 20. Transfer learning.

7.2. Ethical and Safety Considerations in Adversarial Contexts

MADS, while effective in many respects, remains vulnerable to both traditional cyber-threats (such as buffer overflow and Denial-of-Service attacks) and contemporary adversarial machine learning techniques. These include poisoning, evasion, jailbreaks, prompt injection, and model inversion—each of which compromises the CIA of AI systems and challenges existing safety protocols [13,14,29,39,96–98].

1. Relevance of attack types to MADS: Although adversarial ML threats are increasingly prevalent, the current MADS model primarily relies on threshold-based detection and lacks dedicated adversarial defence mechanisms. As such, its behaviour under adversarial conditions remains untested. Deploying such a system in live environments without evaluating its vulnerability could be ethically problematic, risking failure or undetected compromise.
2. Behaviour of MADS against adversarial attacks: The MADS system in its current form does not incorporate adversarial training or robustness optimisation techniques. It is therefore likely to be susceptible to the following:
    - *Data poisoning:* Adversaries may inject crafted false alerts that closely resemble legitimate events, exploiting the limitations of synthetic training data.
    - *Evasion techniques:* Minor feature perturbations may allow adversarial inputs to bypass the model's simple thresholding logic.
    - *Model inversion:* The absence of explainability enables adversaries to probe system outputs and infer internal decision logic.

    Adversarial attacks can target the CIA triad during model training, testing, and deployment. Some examples follow:
    - Confidentiality can be compromised by extracting training data or algorithmic behaviours [99].
    - Integrity may be undermined by altering classification rules, requiring retraining with verified datasets [100].
    - Availability can be disrupted through adversarial reprogramming, resulting in unauthorised actions or system shutdowns [101].
3. Robustification of MADS: To mitigate these vulnerabilities, MADS should incorporate the following:
    - *Adversarial training:* This can be achieved through the use of the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) to strengthen k-NN decision boundaries [102].
    - *Robust statistics:* The is includes the application of methods like the Elliptic Envelope to identify anomalies in the feature space [103].
    - *Differential privacy:* This includes output perturbation strategies to resist inversion and membership inference attacks [104].
    - *Rate-limiting and monitoring:* These provide control to detect and prevent abuse through excessive queries or prompt injections [105].

Robustness testing against adversarial noise and real-world distortions should follow a structured methodology:

1. Generate adversarial examples using FGSM and PGD with varying perturbation strengths; simulate poisoning via mislabelled data.
2. Introduce operational noise (such as Gaussian distortions, packet loss) to mimic sensor faults or logging issues [106].
3. Measure robustness using metrics such as accuracy degradation, FP/FN rates, detection delays, and area under robustness curve (AURC) [107].

4. Benchmark MADS against baseline detectors and apply statistical tests to confirm significant performance deviations.

Continuous monitoring, well-defined break thresholds, and iterative robustness testing should be integrated into the development pipeline. Regular red-team simulations and postmortem analyses using production logs are essential to sustaining resilience in dynamic threat landscapes.

7.2.1. Misuse

While AI significantly enhances industrial and cybersecurity capabilities, it equally empowers malicious actors. TAs now utilise AI to conduct attacks with greater speed, precision, and stealth. By exploiting publicly available APIs and legitimate tools, they can test malware, automate reconnaissance, and identify high-value targets [108]. Generative AI enables the automated crafting of phishing emails, SMS messages, and social engineering content tailored to individual recipients.

These AI-driven campaigns are often fully autonomous and context-aware [109]. In one case, a synthetic voice was used to impersonate an energy company executive, deceiving an employee into transferring approximately USD 250,000 to a fraudulent account [110]. Such misuse underscores the dual-use dilemma of AI technologies.

7.2.2. Limitations

Developing robust AI systems entails significant investment in compute power, memory, and annotated datasets. Many organisations lack the resources to access high-quality data, particularly for rare or sophisticated attack types. Meanwhile, adversaries are accelerating their attack methods by applying neural fuzzing, leveraging neural networks to test vast input combinations and uncover system vulnerabilities [111]. This increasing asymmetry between defence and offence necessitates a more resilient and adaptive AI framework.

7.3. Deployment

The limited knowledge of AI has led to deployment problems, leaving organisations more vulnerable to threats. Certain guiding principles should be applied while deploying AI to ensure support, security, and capabilities availability [112]. This improves the effectiveness, efficiency, and competitiveness, and it should go through a responsible AI framework and planning process together with the current organisational framework to set realistic expectations for AI projects. Some of the guiding principles are as follows [113]:

- Competency: An organisation must have the willingness, resources, and skill set to build home-grown custom AI applications, or a vendor with proven experience in implementing AI-based solutions must be used.
- Data readiness: AI models rely on the quantity and quality of data. But they might be in multiple formats, in different places, or managed by different custodians. Therefore, the use of the data inventory to assess the availability and difficulty of ingesting, cleaning, and harmonising the data is required.
- Experimentation: Implementation is complex and challenging, it requires adoption, fine-tuning and maintenance even with already-built solutions. Experimenting is expected using use cases, learning, and iterating until a successful model is developed and deployed.
- Measurement: An AI system has to be evaluated for its performance and security using a measurement framework. Data must be collected to measure performance and confidence and for metrics.

- Feedback loops: Systems are retrained and evaluated with new data. It is a best practice to plan and build a feedback loop cycle for the model to relearn and to fine-tune it to improve accuracy and efficiency. Workflow should be developed and data pipelines automated to constantly obtain feedback on how the AI system is performing using RL [72].
- Education: Educating the team on the technology's operability is instrumental in having a successful AI deployment, and it will improve efficiency and confidence. The use of AI can help the team grow, develop new skills, and accelerate productivity.

7.4. Product Evaluation

Evaluation is fundamental when acquiring or developing an AI system. Different products are being advertised as AI-enabled with capabilities to detect and prevent attacks, automate tasks, and predict patterns. However, these claims have to be evaluated and identified for scoping and tailoring purposes to fit the organisation's objectives [114]. It is vital to validate processes for model training, applicability, integration, proof of concept, acquisition, support model, reputation, affordability, and security to support practical and reasoned decisions [115].

The documentation of the product's trained model's process, data, duration, accountability, and measures for labelled data unavailability should be provided. It should state if the model has other capabilities, the source of training data, and who will be training the model and managing feedback loops, and a time frame from installation to actionable insights should be provided. A good demonstration of an AI product does not guarantee a successful integration in the environment, and a proof of concept should be developed with enterprise data and in its environment. Any anticipated challenges should be acknowledged and supported. A measurement framework should be developed to obtain meaningful metrics in the ML pipeline. An automated workflow should be developed to orchestrate the testing and deploying of models using a standardised process.

It is important to understand whether AI capabilities were in-built or acquired, that is whether they are an add-on module or part of the underlying product, as well as the level of integration. The security capabilities and features of the product must be evaluated, including data privacy preservation. There should be a clear agreement on the ownership of the data to align with privacy compliance and evaluate vendors' approach and measures to protect the AI system.

8. Conclusions

The fields of AI and cybersecurity are evolving rapidly and can be utilised symbiotically to improve global security. Leveraging AI can yield benefits in defensive security but also empower TAs. This paper discussed the discipline of AI, security objectives, and the applicability of AI in the field of cybersecurity. It presented use cases of ML to solve specific problems and developed a predictive MADS model to demonstrate an AI-enabled detection approach. With proper consideration and preparation, AI can be beneficial to organisations in enhancing security and increasing efficiency and productivity. Overall, AI can improve security operations, vulnerability management, and security posture, accelerate detection and response, and reduce duplication of processes and human fatigue. However, it can also increase vulnerabilities, attacks, violation of privacy, and bias. AI can also be utilised by TAs to initiate sophisticated and stealth attacks. This paper recommended best practices, deployment operation principles, and evaluation processes to enable visibility, explainability, attack surface reduction, and responsible AI. Future work will focus on improving MADS, developing the model's other use cases such as adversarial ML, using real-world samples, and developing a responsible AI evaluation framework for

better accountability, transparency, fairness, and interpretability. This includes testing and quantitatively evaluating the model's robustness against specific adversarial attack vectors.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Zeadally, S.; Adi, E.; Baig, Z.; Khan, I.A. Harnessing artificial intelligence capabilities to improve cybersecurity. *IEEE Access* **2020**, *8*, 23817–23837. [CrossRef]
2. Security, I. The Cost of a Data Breach Report 2022. 2022. Available online: https://community.ibm.com/community/user/events/event-description?CalendarEventKey=7097fd42-4875-4abe-9ff6-d556af01688b&CommunityKey=96f617c5-4f90-4eb0-baec-2d0c4c22ab50&Home=%2Fcommunity%2Fuser%2Fhome (accessed on 10 December 2024).
3. Finance, Y. Artificial Intelligence (AI) in Cybersecurity Market Size USD 102.78 BN by 2032. 2023. Available online: https://www.globenewswire.com/news-release/2023/01/23/2593136/0/en/Artificial-Intelligence-AI-In-Cybersecurity-Market-Size-USD-102-78-BN-by-2032.html (accessed on 15 September 2024).
4. Arp, D.; Quiring, E.; Pendlebury, F.; Warnecke, A.; Pierazzi, F.; Wressnegger, C.; Cavallaro, L.; Rieck, K. Dos and don'ts of machine learning in computer security. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 3971–3988.
5. Williams, J.; King, J.; Smith, B.; Pouriyeh, S.; Shahriar, H.; Li, L. Phishing Prevention Using Defense in Depth. In *Proceedings of the Advances in Security, Networks, and Internet of Things: Proceedings from SAM'20, ICWN'20, ICOMP'20, and ESCS'20*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 101–116.
6. Truong, T.C.; Diep, Q.B.; Zelinka, I. Artificial Intelligence in the Cyber Domain: Offense and Defense. *Symmetry* **2020**, *12*, 410. [CrossRef]
7. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **2021**, *23*, 18. [CrossRef]
8. Nishant, R.; Kennedy, M.; Corbett, J. Artificial intelligence for sustainability: Challenges, opportunities, and a research agenda. *Int. J. Inf. Manag.* **2020**, *53*, 102104. [CrossRef]
9. Wirkuttis, N.; Klein, H. Artificial intelligence in cybersecurity. *Cyber Intell. Secur.* **2017**, *1*, 103–119.
10. Ghillani, D. Deep Learning and Artificial Intelligence Framework to Improve the Cyber Security. *Authorea Prepr.* **2022**, *preprint*. Available online: https://www.authorea.com/users/506161/articles/587142-deep-learning-and-artificial-intelligence-framework-to-improve-the-cyber-security (accessed on 15 May 2025).
11. Tyugu, E. Artificial Intelligence in Cyber Defense. In Proceedings of the 2011 3rd International Conference on Cyber Conflict, Tallinn, Estonia, 7–10 June 2011.
12. Abusnaina, A.; Khormali, A.; Alasmary, H.; Park, J.; Anwar, A.; Mohaisen, A. Adversarial learning attacks on graph-based IoT malware detection systems. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 1296–1305.
13. Chen, T.; Liu, J.; Xiang, Y.; Niu, W.; Tong, E.; Han, Z. Adversarial attack and defense in reinforcement learning-from AI security view. *Cybersecurity* **2019**, *2*, 11. [CrossRef]
14. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw.* **2019**, *30*, 2805–2824. [CrossRef] [PubMed]
15. Li, H.; Guo, Y.; Huo, S.; Hu, H.; Sun, P. Defensive deception framework against reconnaissance attacks in the cloud with deep reinforcement learning. *Sci. China Inf. Sci.* **2022**, *65*, 170305. [CrossRef]
16. Robertson, J.; Fossaceca, J.M.; Bennett, K.W. A cloud-based computing framework for artificial intelligence innovation in support of multidomain operations. *IEEE Trans. Eng. Manag.* **2021**, *69*, 3913–3922. [CrossRef]
17. Pollini, A.; Callari, T.C.; Tedeschi, A.; Ruscio, D.; Save, L.; Chiarugi, F.; Guerri, D. Leveraging human factors in cybersecurity: An integrated methodological approach. *Cogn. Technol. Work* **2022**, *24*, 371–390. [CrossRef]
18. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* **2014**, *41*, 1690–1700. [CrossRef]
19. Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. *Future Gener. Comput. Syst.* **2018**, *82*, 761–768. [CrossRef]
20. Ghorbani, A.A.; Lu, W.; Tavallaee, M. *Network Intrusion Detection and Prevention: Concepts and Techniques*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009; Volume 47.

21. Biggio, B.; Roli, F. Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 2154–2156. [CrossRef]
22. Li, Y.; Ma, R.; Jiao, R. A hybrid malicious code detection method based on deep learning. *Int. J. Secur. Its Appl.* **2015**, *9*, 205–216. [CrossRef]
23. Valdes, A.; Skinner, K. Probabilistic Alert Correlation. In *Recent Advances in Intrusion Detection*; Goos, G., Hartmanis, J., Van Leeuwen, J., Lee, W., Mé, L., Wespi, A., Eds.; Series Title: Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2212, pp. 54–68. [CrossRef]
24. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A deep learning approach to network intrusion detection. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 41–50. [CrossRef]
25. Copeland, B.J. The Turing Test*. *Minds Mach.* **2000**, *10*, 519–539. [CrossRef]
26. Harnad, S. The Turing Test is not a trick: Turing indistinguishability is a scientific criterion. *Acm Sigart Bull.* **1992**, *3*, 9–10. [CrossRef]
27. Kaplan, A.; Haenlein, M. Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Bus. Horiz.* **2019**, *62*, 15–25. [CrossRef]
28. Sommer, R.; Paxson, V. Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 305–316.
29. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *Digit. Threat.* **2022**, *3*, 19. [CrossRef]
30. Chang, V.; Golightly, L.; Modesti, P.; Xu, Q.A.; Doan, L.M.T.; Hall, K.; Boddu, S.; Kobusińska, A. A Survey on Intrusion Detection Systems for Fog and Cloud Computing. *Future Internet* **2022**, *14*, 89. [CrossRef]
31. Moustafa, N.; Koroniotis, N.; Keshk, M.; Zomaya, A.Y.; Tari, Z. Explainable Intrusion Detection for Cyber Defences in the Internet of Things: Opportunities and Solutions. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1775–1807. [CrossRef]
32. Agarwal, B.B.; Tayal, S.P.; Gupta, M. *Software Engineering and Testing*; Jones & Bartlett Learning: Burlington, MA, USA, 2010.
33. Stuttard, D.; Pinto, M. *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
34. John, M.M.; Olsson, H.H.; Bosch, J. Towards mlops: A framework and maturity model. In Proceedings of the IEEE 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Palermo, Italy, 1–3 September 2021; pp. 1–8.
35. Sarker, I.H. Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.* **2021**, *2*, 160. [CrossRef]
36. Mathas, C.M.; Vassilakis, C.; Kolokotronis, N.; Zarakovitis, C.C.; Kourtis, M.A. On the design of IoT security: Analysis of software vulnerabilities for smart grids. *Energies* **2021**, *14*, 2818. [CrossRef]
37. Alenezi, M.; Almuairfi, S. Essential Activities for Secure Software Development. *Int. J. Softw. Eng. Appl* **2020**, *11*, 1–14. [CrossRef]
38. Edris, E.K.K.; Aiash, M.; Loo, J. An Introduction of a Modular Framework for Securing 5G Networks and Beyond. *Network* **2022**, *2*, 419–439. [CrossRef]
39. Kotenko, I.; Saenko, I.; Lauta, O.; Vasiliev, N.; Kribel, K. Attacks Against Artificial Intelligence Systems: Classification, The Threat Model and the Approach to Protection. In Proceedings of the Sixth International Scientific Conference "Intelligent Information Technologies for Industry"(IITI'22), Harbin, China, 25–30 September 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 293–302.
40. Dunsin, D.; Ghanem, M.; Ouazzane, K. The use of artificial intelligence in digital forensics and incident response (DFIR) in a constrained environment. In Proceedings of the International Conference on Digital Forensics and Security of Cloud Computing (ICDFSCC), Sydney, Australia, 17–18 May 2022.
41. Nist. *Cybersecurity Framework CSF*; NIST: Gaithersburg, MD, USA, 2013.
42. Kwon, R.; Ashley, T.; Castleberry, J.; Mckenzie, P.; Gupta Gourisetti, S.N. Cyber Threat Dictionary Using MITRE ATT&CK Matrix and NIST Cybersecurity Framework Mapping. In Proceedings of the 2020 Resilience Week (RWS), Salt Lake City, UT, USA, 19–23 October 2020; pp. 106–112. [CrossRef]
43. Crumpler, W.; Lewis, J.A. *The Cybersecurity Workforce Gap*; JSTOR: New York, NY, USA, 2019.
44. Alshaikh, M. Developing cybersecurity culture to influence employee behavior: A practice perspective. *Comput. Secur.* **2020**, *98*, 102003. [CrossRef]
45. Xiong, W.; Legrand, E.; Åberg, O.; Lagerström, R. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix. *Softw. Syst. Model.* **2022**, *21*, 157–177.
46. Danquah, P. Security Operations Center: A Framework for Automated Triage, Containment and Escalation. *J. Inf. Secur.* **2020**, *11*, 225–240. [CrossRef]

47. Schlette, D.; Caselli, M.; Pernul, G. A comparative study on cyber threat intelligence: The security incident response perspective. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2525–2556. [CrossRef]
48. Cichonski, P.; Millar, T.; Grance, T.; Scarfone, K. *Computer Security Incident Handling Guide*; Technical Report NIST Special Publication (SP) 800-61 Rev. 2; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2012. [CrossRef]
49. Preuveneers, D.; Joosen, W. Sharing machine learning models as indicators of compromise for cyber threat intelligence. *J. Cybersecur. Priv.* **2021**, *1*, 140–163. [CrossRef]
50. Naseer, H.; Maynard, S.B.; Desouza, K.C. Demystifying analytical information processing capability: The case of cybersecurity incident response. *Decis. Support Syst.* **2021**, *143*, 113476. [CrossRef]
51. Salitin, M.A.; Zolait, A.H. The role of User Entity Behavior Analytics to detect network attacks in real time. In Proceedings of the IEEE 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 18–20 November 2018; pp. 1–5.
52. Kumar, J.; Santhanavijayan, A.; Rajendran, B. Cross site scripting attacks classification using convolutional neural network. In Proceedings of the IEEE 2022 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 25–27 January 2022; pp. 1–6.
53. McHugh, J. Intrusion and intrusion detection. *Int. J. Inf. Secur.* **2001**, *1*, 14–35. [CrossRef]
54. Gassais, R.; Ezzati-Jivan, N.; Fernandez, J.M.; Aloise, D.; Dagenais, M.R. Multi-level host-based intrusion detection system for Internet of things. *J. Cloud Comput.* **2020**, *9*, 1–16. [CrossRef]
55. Saxena, N.; Hayes, E.; Bertino, E.; Ojo, P.; Choo, K.K.R.; Burnap, P. Impact and key challenges of insider threats on organizations and critical businesses. *Electronics* **2020**, *9*, 1460. [CrossRef]
56. Ahmed, H.; Traore, I.; Saad, S.; Mamun, M. Automated detection of unstructured context-dependent sensitive information using deep learning. *Internet Things* **2021**, *16*, 100444. [CrossRef]
57. Agrawal, E.G.; Goyal, S.J. Survey on Data Leakage Prevention through Machine Learning Algorithms. In Proceedings of the IEEE 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 10–11 March 2022; pp. 121–123.
58. Ghouse, M.; Nene, M.J.; Vembuselvi, C. Data leakage prevention for data in transit using artificial intelligence and encryption techniques. In Proceedings of the IEEE 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 20–21 December 2019; pp. 1–6.
59. Chung, M.H.; Yang, Y.; Wang, L.; Cento, G.; Jerath, K.; Raman, A.; Lie, D.; Chignell, M.H. Implementing Data Exfiltration Defense in Situ: A Survey of Countermeasures and Human Involvement. *Acm Comput. Surv.* **2023**, *55*, 1–37. [CrossRef]
60. Reeves, A.; Delfabbro, P.; Calic, D. Encouraging employee engagement with cybersecurity: How to tackle cyber fatigue. *SAGE Open* **2021**, *11*, 21582440211000049. [CrossRef]
61. Yang, Y.; Zheng, X.; Guo, W.; Liu, X.; Chang, V. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Inf. Sci.* **2019**, *479*, 567–592. [CrossRef]
62. Eunaicy, J.C.; Suguna, S. Web attack detection using deep learning models. *Mater. Today Proc.* **2022**, *62*, 4806–4813. [CrossRef]
63. Yüksel, O.; den Hartog, J.; Etalle, S. Reading between the fields: Practical, effective intrusion detection for industrial control systems. In Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, 4–8 April 2016; pp. 2063–2070.
64. Williams, R.; Yampolskiy, R. Understanding and Avoiding AI Failures: A Practical Guide. *Philosophies* **2021**, *6*, 53. [CrossRef]
65. Nathezhtha, T.; Sangeetha, D.; Vaidehi, V. WC-PAD: Web crawling based phishing attack detection. In Proceedings of the IEEE 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–6.
66. Assefa, A.; Katarya, R. Intelligent phishing website detection using deep learning. In Proceedings of the IEEE 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 25–26 March 2022; Volume 1, pp. 1741–1745.
67. Sharma, A.; Gupta, B.B.; Singh, A.K.; Saraswat, V.K. Orchestration of APT malware evasive manoeuvers employed for eluding anti-virus and sandbox defense. *Comput. Secur.* **2022**, *115*, 102627. [CrossRef]
68. Gupta, I.; Gupta, R.; Singh, A.K.; Buyya, R. MLPAM: A machine learning and probabilistic analysis based model for preserving security and privacy in cloud environment. *IEEE Syst. J.* **2020**, *15*, 4248–4259. [CrossRef]
69. Ray, S. A quick review of machine learning algorithms. In Proceedings of the IEEE 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 35–39.
70. Sharma, N.; Sharma, R.; Jindal, N. Machine learning and deep learning applications-a vision. *Glob. Transit. Proc.* **2021**, *2*, 24–28. [CrossRef]
71. Arboretti, R.; Ceccato, R.; Pegoraro, L.; Salmaso, L. Design of Experiments and machine learning for product innovation: A systematic literature review. *Qual. Reliab. Eng. Int.* **2022**, *38*, 1131–1156. [CrossRef]
72. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
73. Yoon, J.; Drumright, L.N.; Van Der Schaar, M. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2378–2388. [CrossRef]

74. Panesar, A.; Panesar, A. Machine learning algorithms. In *Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 85–144.
75. Paper, D.; Paper, D. Predictive Modeling Through Regression. In *Hands-on Scikit-Learn for Machine Learning Applications: Data Science Fundamentals with Python*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 105–136.
76. VanDam, C.; Masrour, F.; Tan, P.N.; Wilson, T. You have been caute! early detection of compromised accounts on social media. In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Vancouver, BC, Canada, 27–30 August 2019; pp. 25–32.
77. Sambangi, S.; Gondi, L. A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression. *Proceedings* **2020**, *63*, 51. [CrossRef]
78. Jain, A.K.; Dubes, R.C. *Algorithms for Clustering Data*; Prentice Hall: Englewood Cliffs, NJ, USA, 1988.
79. Ezugwu, A.E.; Ikotun, A.M.; Oyelade, O.O.; Abualigah, L.; Agushaka, J.O.; Eke, C.I.; Akinyelu, A.A. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104743. [CrossRef]
80. Adolfsson, A.; Ackerman, M.; Brownstein, N.C. To cluster, or not to cluster: An analysis of clusterability methods. *Pattern Recognit.* **2019**, *88*, 13–26. [CrossRef]
81. de Melo, C.M.; Torralba, A.; Guibas, L.; DiCarlo, J.; Chellappa, R.; Hodgins, J. Next-generation deep learning based on simulators and synthetic data. In *Trends in Cognitive Sciences*; Elsevier: Amsterdam, The Netherlands, 2021.
82. Wan, Z.; Hazel, J.W.; Clayton, E.W.; Vorobeychik, Y.; Kantarcioglu, M.; Malin, B.A. Sociotechnical safeguards for genomic data privacy. *Nat. Rev. Genet.* **2022**, *23*, 429–445. [CrossRef] [PubMed]
83. Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; Wang, F.Y. Generative adversarial networks: Introduction and outlook. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 588–598. [CrossRef]
84. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
85. Sudhakar, K.N.; Shanthi, M. Deepfake: An Endanger to Cyber Security. In Proceedings of the 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 11–16 June 2023; pp. 1542–1548. [CrossRef]
86. Azab, A.; Khasawneh, M. MSIC: Malware Spectrogram Image Classification. *IEEE Access* **2020**, *8*, 102007–102021. [CrossRef]
87. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223; ISSN 2640-3498.
88. NIST. *Artificial Intelligence Risk Management Framework (AI RMF 1.0);* NIST: Gaithersburg, MD, USA, 2023.
89. *ISO/IEC 42001:2023*; Information Technology—Artificial intelligence—Management System. ISO: Geneva, Switzerland, 2023.
90. Bouacida, N.; Mohapatra, P. Vulnerabilities in federated learning. *IEEE Access* **2021**, *9*, 63229–63249. [CrossRef]
91. Deshmukh, A.A.; Hundekari, S.; Dongre, Y.; Wanjale, K.; Maral, V.B.; Bhaturkar, D. Explainable AI for Adversarial Machine Learning: Enhancing Transparency and Trust in Cyber Security. *J. Electr. Syst.* **2024**, *20* . [CrossRef]
92. Kumar, P.; Gupta, G.P.; Tripathi, R. An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks. *Comput. Commun.* **2021**, *166*, 110–124. [CrossRef]
93. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]
94. Gao, Q.; Luo, Z.; Klabjan, D.; Zhang, F. Efficient architecture search for continual learning. *IEEE Trans. Neural Networks Learn. Syst.* **2022**, *34*, 8555–8565. [CrossRef]
95. Susan, S.; Kumar, A. The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art. *Eng. Rep.* **2021**, *3*, e12298. [CrossRef]
96. Enisa. Securing Machine Learning Algorithms. 2021. Available online: https://www.enisa.europa.eu/publications/securing-machine-learning-algorithms (accessed on 20 December 2024).
97. Ghimire, S.; Thapaliya, S. Al-Driven Cybersecurity: Mitigating Prompt Injection Attacks through Adversarial Machine Learning. *Nprc J. Multidiscip. Res.* **2024**, *8*, 63–69. [CrossRef]
98. Liu, Y.; Yang, C.; Li, D.; Ding, J.; Jiang, T. Defense Against Adversarial Attacks on No-Reference Image Quality Models with Gradient Norm Regularization. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024. [CrossRef]
99. Zhao, B.Z.H.; Agrawal, A.; Coburn, C.; Asghar, H.J.; Bhaskar, R.; Kaafar, M.A.; Webb, D.; Dickinson, P. On the (in) feasibility of attribute inference attacks on machine learning models. In Proceedings of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 6–10 September 2021; pp. 232–251.
100. Ala-Pietilä, P.; Bonnet, Y.; Bergmann, U.; Bielikova, M.; Bonefeld-Dahl, C.; Bauer, W.; Bouarfa, L.; Chatila, R.; Coeckelbergh, M.; Dignum, V. *The Assessment List for Trustworthy Artificial Intelligence (ALTAI)*; European Commission: Brussel, Belgium, 2020.

101. Apruzzese, G.; Colajanni, M.; Ferretti, L.; Marchetti, M. Addressing adversarial attacks against security systems based on machine learning. In Proceedings of the IEEE 2019 11th International Conference on Cyber Conflict (CyCon), Tallinn, Estonia, 28–31 May 2019; Volume 900, pp. 1–18.
102. Villegas-Ch, W.; Jaramillo-Alcázar, A.; Luján-Mora, S. Evaluating the robustness of deep learning models against adversarial attacks: An analysis with fgsm, pgd and cw. *Big Data Cogn. Comput.* **2024**, *8*, 8. [CrossRef]
103. Gu, Q.; Fallah, A.; Ashok, P.; Chen, D.; Van Oort, E. Real-Time Multi-Event Anomaly Detection using Elliptic Envelope and A Deep Neural Network for Enhanced MPD Robustness. In Proceedings of the SPE/IADC Managed Pressure Drilling and Underbalanced Operations Conference and Exhibition, Virtual, 14–16 September 2021; p. D032S011R002.
104. Bai, L.; Hu, H.; Ye, Q.; Li, H.; Wang, L.; Xu, J. Membership Inference Attacks and Defenses in Federated Learning: A Survey. *ACM Comput. Surv.* **2025**, *57*, 1–35. [CrossRef]
105. Jones, N.; Whaiduzzaman, M.; Jan, T.; Adel, A.; Alazab, A.; Alkreisat, A. A CIA Triad-Based Taxonomy of Prompt Attacks on Large Language Models. *Future Internet* **2025**, *17*, 113. [CrossRef]
106. Li, C.; Xu, M.; Du, Y.; Liu, L.; Shi, C.; Wang, Y.; Liu, H.; Chen, Y. Practical Adversarial Attack on WiFi Sensing Through Unnoticeable Communication Packet Perturbation. In Proceedings of the 30th Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 30 September–4 October 2024; ACM MobiCom'24, pp. 373–387. [CrossRef]
107. Zhang, X.Y.; Xie, G.S.; Li, X.; Mei, T.; Liu, C.L. A Survey on Learning to Reject. *Proc. IEEE* **2023**, *111*, 185–215. [CrossRef]
108. Wong, A.Y.; Chekole, E.G.; Ochoa, M.; Zhou, J. On the Security of Containers: Threat Modeling, Attack Analysis, and Mitigation Strategies. *Comput. Secur.* **2023**, *128*, 103140. [CrossRef]
109. Marino, D.L.; Wickramasinghe, C.S.; Manic, M. An Adversarial Approach for Explainable AI in Intrusion Detection Systems. In Proceedings of the IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 21–23 October 2018; pp. 3237–3243. [CrossRef]
110. Manyam, S. Artificial Intelligence's Impact on Social Engineering Attacks 2022. Available online: https://opus.govst.edu/cgi/viewcontent.cgi?article=1521&context=capstones (accessed on 5 January 2025).
111. Wang, Y.; Wu, Z.; Wei, Q.; Wang, Q. Neufuzz: Efficient fuzzing with deep neural network. *IEEE Access* **2019**, *7*, 36340–36352. [CrossRef]
112. Sjödin, D.; Parida, V.; Palmié, M.; Wincent, J. How AI capabilities enable business model innovation: Scaling AI through co-evolutionary processes and feedback loops. *J. Bus. Res.* **2021**, *134*, 574–587. [CrossRef]
113. Uren, V.; Edwards, J.S. Technology readiness and the organizational journey towards AI adoption: An empirical study. *Int. J. Inf. Manag.* **2023**, *68*, 102588. [CrossRef]
114. Corrigan, C.C. Lessons learned from co-governance approaches–Developing effective AI policy in Europe. In *The 2021 Yearbook of the Digital Ethics Lab*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 25–46.
115. Lwakatare, L.E.; Crnkovic, I.; Bosch, J. DevOps for AI–Challenges in Development of AI-enabled Applications. In Proceedings of the IEEE 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Hvar, Croatia, 17–19 September 2020; pp. 1–6.